КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Навчально-науковий інститут філології
Кафедра української мови та прикладної лінгвістики

Автоматичне визначення рівня CEFR українськомовних текстів та створення рівневого за шкалою CEFR корпусу текстів УМІ

Кваліфікаційна робота бакалавра
студента 4 курсу
Освітньої програми
Прикладна (комп'ютерна) лінгвістика та англійська мова"
спеціальності – 035.10 Філологія
(прикладна лінгвістика)
галузі знань – 03 гуманітарні науки
Кійохіко-Кирила ЯСУДИ
Науковий керівник:
к. т. н., доц. Микола КОСТІКОВ

Допущено до захисту
Протокол засідання кафедри української мови та прикладної лінгвістики
№ 15 від «06» 06 2024 року
Завідувач кафедри к. філол. н. доц. Сергій РІЗНИК

Київ — 2024

Зміст

Перелік термінів та скорочень

Термін або скорочення	***Тлумачення***
CEFR	Common European Framework of Reference (укр. Загальноєвропейські рекомендації з мовної освіти)
УМІ	Українська мова як іноземна
NLP	Natural language processing (укр. Обробка природної мови)

Вступ

Створення лінгводидактичних матеріалів має неабияку потребу для української мови як іноземної. З піднесенням популярності вивчення української мови серед іноземців (у тому числі іноземних студентів) з'являється потреба у диференціюванні текстів та створенні завдань за різними рівнями компетенцій володіння мовою. Для цього для української мови було укладено стандарт для кожного рівня CEFR [2; 17]. Втім, лінгводидактичні матеріали не встигають за нововведенням: як у відкритому доступі, так і комерційні матеріали за рівнями CEFR існують у дуже малій кількості. Ця робота покликана розробити допоміжний матеріал, що може бути корисним при диференціації текстів за рівнем складності та потенційне створення завдань на їх основі.

У минулій роботі зі спорідненою темою [1; 18] було проведено експеримент зі спробою написання моделі, що буде здатна класифікувати тексти українською мовою за рівнями CEFR [2; 17]. Втім, результати показали вкрай малу ефективність даної моделі, що була нижче рівня вгадування. Основними причинами були:

- мала кількість тренувального матеріалу
    - цілком пов'язано з специфікою української мови, що є низькоресурсною станом на зараз
- використання керованого алгоритму навчання Multinomial Naive Bayes
    - примітивність даного алгоритму не була здатна врахувати лінгвістичні фактори і лише враховувала статистичні важелі при класифікації

Саме тому в цій роботі було поставлено інший вектор дослідження: фокус

поставлено на способах створення рівневого корпусу за допомогою використання комбінації алгоритмів. За успішних умов вирішення цієї задачі, проблема низькоресурсності української мови буде менш релевантна.

Метою цієї роботи є створення та тестування існуючих алгоритмів для текстової класифікації для задачі класифікації текстів українською мовою за рівнями CEFR.

Завдання, поставлені у цій роботі, наступні:

1. Дослідження існуючих алгоритмів класифікації текстів
2. Написання алгоритму для моделі класифікації текстів за рівнями CEFR
   2.1. Використання існуючих алгоритмів та індексів
   2.2. Створення власного індексу на основі морфологічної розмітки
3. Створення програми, що аналізує тексти на рівень CEFR та укладає корпус
4. Створення графічного інтерфейсу користувача для даної програми
5. Аналіз результатів

Об’єктом дослідження у даній роботі виступають тексти, написані українською мовою.

Предметом дослідження є потенціал автоматичної класифікації цих текстів за рівнями CEFR.

Актуальність даної роботи обумовлена як зростанням популярності української мови серед людей, що вивчають мову, так і через потенційні (у випадку результативності) оптимізації у сфері лінгводидактики. Серед них: автоматичний підбір текстів для опрацювання за рівнями учнів УМІ, автоматичне створення словників та задач УМІ за рівнями на основі рівневих корпусів, потенційна можливість оптимізувати роботу присвоєння мовцеві рівня володіння УМІ.

Розділ 1 Теоретичні засади поставленої задачі

Задача визначення рівня тексту підпадає під загальну задачу класифікації тексту у комп'ютерній лінгвістиці та машинному навчанні загалом. Ця задача полягає у тому, щоб побудувати такий алгоритм, що буде мати здатність на ввід отримати певний текст, обробити його, і на виході отримати один з класів (із попередньо визначеного переліку), у який модель поставила цей текст.

Втім, задача класифікації тексту зазвичай має прямолінійний спосіб вирішення частково або повністю з використанням методів машинного навчання за умови наявності ресурсів на створення набору тренувальних і тестувальних даних для моделі. У даному ж випадку набору готових проанотованих даних для створення моделі не існує; це можна пояснити як низькоресурсністю української мови в інтернет-просторах, так і обмеженим науково-технічним інтересом до даної теми.

Однак способи вирішення задачі визначення рівня CEFR тексту були представлені у наукових працях, але для інших мов [6; 13; 20; 27]. Детальний їх огляд представлено у підрозділі 1.2. Втім варто зазначити, що в більшості випадків ця міра була додатковою до основних підходів машинного навчання у зв'язку з наявністю тренувальних даних та намірами покращити модель за допомогою додаткових параметрів.

1.1 Граматичні вимоги володіння УМІ за стандартом

Стандартизовані вимоги до рівнів володіння українською мовою як іноземною A1-C2 [17] (далі — Стандарт)[1] визначає вимоги до володіння

1 Ця робота ґрунтується на затвердженому стандарті 2018 року. Протягом написання цієї роботи тривали громадські обговорення та розробка нового проєкту вимог [15], але у зв'язку з паралельністю цих процесів огляд проєкту 2024 року та ґрунтування на ньому у роботі представлено не буде.

рівнями УМІ, включно з граматичними, лексичними та тематичними аспектами, якими має володіти студент УМІ на кожному рівні. Конспективний огляд очікуваних компетенцій кожного рівня стандарту наведено в Табл. 1. Українська мова як іноземна визначається як українська мова, опановувана неносіями.

Запровадження в Україні стандарту надає викладачам української мови для іноземців у всьому світі надійну основу для розробки чітких і узгоджених навчальних програм, видання нових сучасних підручників і словників, а також для відкриття україномовних програм та мовних центрів. Завдяки цьому документу діяльність викладачів української мови як іноземної та центрів україністики по всьому світу ставатиме більш систематизованою і координованою.

Стандарт встановлює на державному рівні єдину уніфіковану систему визначення рівнів володіння мовою. Документ містить чіткий перелік комунікативних умінь і навичок для кожного рівня володіння (від A1 до C2) за всіма видами мовленнєвої діяльності, включаючи слухання, читання, письмо та говоріння. Ці вимоги докладно описані відповідно до Загальноєвропейських рекомендацій з мовної освіти.

Стандартизовані вимоги значно впливають на якість і структуру навчального процесу. Вони надають можливість викладачам чітко розуміти, які знання і навички повинні бути сформовані у студентів на кожному рівні. Це сприяє створенню навчальних матеріалів, які відповідають міжнародним стандартам і є адаптованими до специфічних потреб іноземних студентів, що вивчають українську мову.

Табл. 1: Конспективний огляд рівнів за стандартом

Рівень за CEFR:	***A1***	***A2***	***B1***	***B2***	***C1***	***C2***
Українська назва рівня:	Початковий	Базовий	Рубіжний	Середній	Високий	Вільне володіння
Опис:	Часткове ознайомлення зі структурою і системою української мови	Основні опорні риси володіння українською мовою як іноземною	Необхідний рівень володіння українською мовою як іноземною	Достатній рівень володіння українською мовою як іноземною	Активне володіння українською мовою як іноземною	Вільне володіння українською мовою як іноземною
Очікувана кількість опанованої лексики (у лексемах):	800-1000	2000	3000-4000	5000-6000	7000-8000	9000-10000

Документ стандарту також надає каталоги з конкретнішими аспектами компетенцій для кожного рівня. Каталоги А містять переліки комунікативних ролей, каталоги Б — переліки тематик, які мовець повинен вміти обговорювати та розуміти і каталоги В, найбільш корисні для даного дослідження та формально деталізовані — переліки мовних компетенцій, що включають у себе фонетичні, орфографічні, граматичні компетенції для рівнів.

Каталоги В, зокрема підрозділи, що стосуються компетенцій у морфології, було

взято за основу для розробки граматичного індексу, який використовується в даному дослідженні. Ці каталоги містять детальний перелік граматичних структур та словотворчих елементів, що є ключовими для оцінки рівня володіння мовою. Використання таких каталогів забезпечує систематичний підхід до аналізу мовних компетенцій, дозволяючи створити індекс, який точно відображає рівень мовної підготовки.

Процес використання та модифікації каталогів B для формалізації параметрів задля їх подальшої машинної обробки докладно описано в підрозділі 2.2.2 цього дослідження. У цьому підрозділі розглянуто методологічні підходи до адаптації існуючих лінгвістичних ресурсів для потреб автоматизованого аналізу. Зокрема, розроблено процедури для переведення описових лінгвістичних характеристик у формат, придатний для алгоритмічного опрацювання. Це включає створення анотованих баз даних текстів, де кожен мовний елемент марковано відповідно до встановлених каталогів, а також розробку алгоритмів, здатних ідентифікувати та класифікувати ці елементи в автоматичному режимі.

Сам стандарт заснований на Загальноєвропейських рекомендаціях з мовної освіти [2]. Рекомендації широко застосовані для мов Європи та поза нею, та використовуються для диференціації тих, хто вивчає мову, на основі їхніх здібностей. Стандарти окреслюють загальні комунікативні вміння для кожного рівня, такі як спитати шлях або перепросити. Згодом Асоціацією мовного тестування Європи був створений спрощений опитувальник близько 400+ питань формату «можу зробити» для визначення рівня на основі рекомендацій, базуючись на комунікативних намірах. Втім, як зазначено в коментарі [3, с. 391], «імплементація рекомендацій має проблеми в сфері неврахування особливостей граматичних аспектів кожної окремо взятої мови: наприклад,

здатність пояснити місце свого проживання може бути синтаксично простою в одній мові, і водночас морфологічно комплексною в іншій». Для стандартизованих вимог УМІ також є ряд зауважень від лінгвістичної спільноти [16].

1.2 Огляд подібних минулих досліджень

У зв’язку з незадовільними результатами моделей машинного навчання або відсутністю відповідних корпусів, дослідники використовували та створювали індекси для допомоги або заміни потреби в алгоритмах. Ці індекси розраховують різноманітні лінгвістичні та текстові параметри, а також відношення між ними, і здійснюють шкалювання, забезпечуючи альтернативні методи оцінювання мовних характеристик текстів [8; 27]. Окрім задач класифікації рівня CEFR цей підхід також застосовувався у задачах автоматичного оцінювання письмових робіт на предмет читабельності та «якісності» [8].

У 2012 для французької мови як іноземної було опрацьовано датасет, додавши до нього близько 46 текстових параметрів для подальшої обробки моделями машинного навчання (автори статті зупинилися на моделі SVM, що показала найкращі результати) [20]. Перевага, яку мали дослідники — наявність «золотого стандарту» для даної задачі — наявність корпусів текстів з маркуванням рівнів CEFR [20]. Було використано лексичні індекси, такі як класичні метрики середнього значення слів на речення та літер на слово (Це у свою чергу привело до висновків, що тексти, узяті з підручників для рівнів A1-B1, по своїй формі досить гетерогенні та мали суб’єктивне сприйняття рівнів укладачами), граматичні індекси, а також семантичні індекси, які показали себе неефективними для даної задачі [20].

В іншій статті 2020 року дослідниками було опрацьовано тексти італійської мови як іноземної для аналогічної задачі класифікації «складності» тексту (англ. *Text complexity*) [13]. Так само як і в минулій описаній роботі, була спроба ввести додаткові параметри для тренування моделі. Корпус текстів був представлений як набір n-вимірних векторів, де кожен вектор — текст, виміри вектора — лінгвістичні параметри, що потенційно здатні дискримінувати комплексність текстів. Усього було використано та протестовано 139 квантитативних параметрів [13]. При обробці даних не було додано семантичних параметрів, а також вектори текстів не були векторами-ембедингами, тобто не мали інформації про математичну репрезентацію відношень значення між словами в текстах. Також рівні A1, A2 було вилучено з розмітки та датасету через неможливість їхньої диференціації.

Інша стаття 2021 року [19], де було оброблено англійську мову як іноземну за рівнями, мала 57 лінгвістичних та текстових параметрів, що відносилися до 4 категорій:

1. параметри синтаксичної складності
2. параметри лексичного багатства
3. виміри словосполучень
4. інформаційно-теоретичні виміри

Дослідники називають цей підхід до створення додатковий параметрів «контурами комплексності» (англ. *Complexity contours*) [19]. Метод контурів комплексності детально описано в [22].

Втім, як і минулі рази, дані параметри радше слугували допоміжними параметрами при тренуванні моделі машинного навчання. Автори використовували корпус EFCAMDAT [7], розмічений корпус текстів студентів-

іноземців англійською мовою, де кожен текст має маркування рівня CEFR та інші анотації всередині текстів. Наявність цього «золотого стандарту» у вигляді власне існуючих розміток робить лінгвістичні параметри вторинними, хоча й важливими, аспектами досліження.

Досвід даної задачі для українськомовних текстів досить малий, та обмежується загальними параметрами, без прив'язки до задач лінгводидактики.

1.3 Індекси, метрики, параметри

Протягом огляду минулих робіт було виявлено декілька індексів, які буде використано в даному дослідженні для виміру «складності» та «читабельності» тексту.

У даному дослідженні використано Automated Readability Index (укр. *Автоматизований індекс читабельності*) (далі — ARI). ARI – індекс «читабельності», що був створений американськими дослідниками під егідою американських оборонних відомств у 1967 році задля моніторингу читабельності на друкарських машинках у режимі реального часу. Він пропонував автоматизований метод підрахунку [14]. Користь індекса досліджувалась в умовах військових тренувань американського морського флоту: була необхідність розуміння й оцінки «читабельності» видаткових навчальних матеріалів для новоприбулих, тобто того, наскільки матеріал був зрозумілим для прочитання і вивчення [9]. Окрім цього у документі [9] йдеться також про коригування під військові потреби інших індексів, таких як індекс Флеша (згодом індекс Флеша-Кінкейда, див. ст. 14, а також [9]), індекс туманності Ганнінга, тощо; Усі формули, які було обрано, містили параметр речення. Вони є більш точними для технічного матеріалу.

Посібники були набагато вищими за рівень читання персоналу. Начальник

військово-морської технічної підготовки назвав це давньою проблемою. Дослідження також показало відмінності між формулами. ARI виявилась більш надійною для їх використання, ніж індекс Флеша [9].

Це пояснюється тим, що автоматизація підвищила точність. На той час вона пропонувала більш зручне рішення.

Індекс ARI визначений наступним розрахунком:

$$\mathrm{ARI} = 4.71 \times {C}/{W} + 0.5 \times {W}/{S} - 21.43$$

де:

- C : кількість символів у тексті
- W : кількість слів у тексті
- S : кількість речень у тексті

На вихід розраховується індекс, інтерпретацію якого подано на Табл. 2 відповідно до американської системи освітніх рівнів; очікувалося, що маючи рівень освіти, що співвідноситься з індексом, читач здатен зрозуміти текст.

Табл. 2: Відповідність рівнів індексу ARI з рівнями американської системи освіти

Значення індексу ARI	***Рівень читання***
1	Дитячий садок
2	1-2 класи
3	3 клас
4	4 клас
5	5 клас
6	6 клас

7	7 клас
8	8 клас
9	9 клас
10	10 клас
11	11 клас
12	12 клас
13	Студент ВНЗ
14	Професор

Також у даному дослідженні використано індекс рівня Флеша-Кінкейда [9]. Він розраховується наступним чином:

$$\text{Flesch Kincaid} = 0.39 \times {W}/{Sents} + 11.8 \times {Sylls}/{W} - 15.59$$

де:

- W : кількість слів у тексті
- $Sents$: кількість речень у тексті
- $Sylls$: сумарна кількість складів у словах у тексті

На вихід видається індекс рівня Флеша-Кінкейда, еквівалентний американському рівню освіти. Він показує рівень освіти, необхідний для розуміння тексту. Інтерпретацію даного індексу наведено на Табл. 3.

Табл. 3: Інтерпретація індексу рівня Флеша-Кінкейда

Значення індексу	***Рівень освіти***
0-3	Дитячий садок / Початкова школа

Значення індексу	***Рівень освіти***
3-6	Початкові класи
6-9	Середня школа
9-12	Старша школа
12-15	Коледж / Університет
15-18	Аспірантура / Післядипломна освіта

Індекс Флеша-Кінкейда є найбільш поширеною у різних сферах в Америці, від маркетингу до державного управління. У той час як деякі формули є більш спеціалізованими, метод Флеша-Кінкейда є найбільш рекомендованим для всіх галузей і дисциплін [5].

Його також розробили військово-морські сили США, які працювали з індексом Flesch Reading Ease. Раніше бали за шкалою Flesch Reading Ease потрібно було перераховувати за допомогою таблиці, щоб перевести їх у бали за рівнем читання. Змінена версія була розроблена в 1970-х роках, щоб зробити її простішою у використанні. Військово-морський флот використовував її для своїх технічних посібників, що використовувалися під час навчання [9].

Нові формули читабельності були розроблені за допомогою регресійного аналізу [9; 10]. Для розрахунку цих формул було залучено 569 інформантів, усі з яких були новоприбулими американськими військовослужбовцями (з терміном служби до 6 місяців на момент проведення дослідження). Переважна більшість учасників дослідження були випускниками старшої школи. З метою дослідження їм було запропоновано пройти тести на читання та розуміння текстового матеріалу.

Результати тестування дозволили зібрати фактичні дані про рівень читабельності текстів серед інформантів. На основі цих даних формули читабельності були уточнені та скориговані за допомогою методів регресійного аналізу, що дозволило підвищити точність і надійність отриманих результатів. Зокрема, регресійний аналіз допоміг виявити ключові фактори, які впливають на читабельність текстів, і врахувати їх у нових формулах.

У висновку, існуючі індекси «читабельності» широко використовувані і під час задач ідентифікації рівня CEFR, що підтверджено успішними випадками тренування моделі машинного навчання на цих параметрах для класифікації рівнів CEFR різних європейських мов.

Розділ 2 Пропонований процес автоматичного визначення рівня тексту УМІ за шкалою CEFR

Для роботи над задачею ідентифікації рівня CEFR українськомовного тексту було використано високорівневу мову програмування Python версії 3.8.0 [24]. IDE, на якому було написано програмний код — PyCharm 2024.1.1 [12].

Для написання графічного користувацького інтерфейсу було використано бібліотеку PyQt6 [21].

Для створення таблиць, їх модифікації, обробки табличних даних було використано бібліотеку pandas [23; 25].

Для створення та обробки морфологічної розмітки було використано бібліотеку pymorphy2 [11].

Програма складається з кількох файлів з програмним кодом у форматі `.py`, усі файли наведено в додатках. Основні компоненти програми включають наступні файли: `cefr_check_utils.py`, `readability_utils.py`, `data_utils.py`, і `main.py`.

Файли `cefr_check_utils.py`, `readability_utils.py` та `data_utils.py` містять набір функцій, які відповідають за обробку даних та розрахунок різноманітних індексів. Зокрема, `cefr_check_utils.py` включає функції, що перевіряють відповідність текстів рівням загальноєвропейських рекомендацій з мовної освіти (CEFR) за індексами граматичних компетентностей УМІ (Див. підрозділ 2.2.2). У файлі `readability_utils.py` зосереджено функції, які розраховують показники читабельності текстів, враховуючи різні лінгвістичні параметри. Файл `data_utils.py` містить утиліти для обробки та підготовки даних, необхідних для подальших аналізів.

Головним файлом програмного коду є `main.py`. Цей файл відповідає за створення та управління графічним інтерфейсом користувача (GUI), який дозволяє користувачам взаємодіяти з програмою в зручний та інтуїтивно зрозумілий спосіб. У `main.py` також здійснюється інтеграція функцій з інших файлів, що забезпечує їх спільну роботу в рамках єдиного додатка. Файл містить виклики до розрахункових функцій, реалізованих у `cefr_check_utils.py`, `readability_utils.py`, і `data_utils.py`, що дозволяє виконувати аналізи текстів на основі оброблених даних.

Детальніше про допоміжні файли коду див. ст. 19.

2.1 Набір та обробка даних

На вхід написана програма приймає текстовий файл формату `.txt`.

Після цього програма створює дві таблиці на основі тексту:

1. Таблиця речень: текст, токенізований на речення, а також текстовий параметр кількості слів у кожному окремо взятому реченні.
2. Таблиця слів: текст, токенізований на слова (токени), а також параметри:
    2.1. Лематизована форма кожного токена
    2.2. Морфологічні теги кожного токена
    2.3. Довжина (символьна) токена
    2.4. Довжина (складо́ва) токена

Код, що означує функції для створення таблиць, наведено у додатках (див. Додаток 2).

2.2 Принцип роботи програми

У зв’язку з відсутністю будь-яких корпусів текстів українською мовою, проанотованих за рівнями CEFR, було прийняте рішення скористатися методами, що мають мінімальне або нульове залучення підходів машинного навчання, що передбачають тренування, валідацію та тестування моделі на вичерпній кількості маркованих даних.

Власне послідовність дій роботи програми наступна:

1) Користувач задає текст українською мовою у форматі `.txt`
2) Програма обробляє його виконуючи:
   2.1) токенізацію тексту на речення та слова
   2.2) здійснення параметризації слів та речень за кількісними показниками
   2.3) обчислення індексів читабельності ARI та рівневий індекс Флеша-Кінкейда
   2.4) виведення у графічний інтерфейс користувача звіту про автоматичний аналіз рівня CEFR тексту
3) Користувач зберігає звіт та корпуси речень і слів за вказаною адресою розташування

Програма використовує наступні бібліотеки, створені для потреб цього дослідження або багатофункціональні:

1. cefr_check_utils (Додаток 3): для перевірки рівня грамем за CEFR (Common European Framework of Reference for Languages).
2. data_utils (Додаток 2): для обробки даних тексту, створення таблиць слів

та речень.

3. readability_utils (Додаток 4): для обчислення індексів читабельності.
4. Pandas [23; 25]: для обробки і збереження таблиць даних.
5. Datetime [4]: для отримання поточної дати.
6. PyQt6.QtWidgets [21]: для створення елементів графічного інтерфейсу користувача.

2.2.1 Індекси читабельності

Для даної програми було прийняте рішення залучити індекс ARI та індекс рівня Флеша-Кінкейда [9].

Дані індекси було імплементовано в коді шляхом означення наступних функцій:

```
001.  def ARI_index(sent_count, word_count, char_count):
002.      '''
003.        Calculate ARI index from sentence count and word count and
   character count.
004.      :param sent_count: integer, total number of sentences
005.      :param word_count: integer, total number of words
006.      :param char_count: integer, total number of characters
007.      :return: integer, ARI index
008.      '''
009.       return 4.71 * (char_count / word_count) + 0.5 * (word_count /
   sent_count) - 21.43
010.
011.  def       Flesh_Kincaid_grade_level(sentence_count,       word_count,
   syllable_count):
012.      '''
013.      Calculate Flesh Kincaid grade level.
```

```
014.       :param sentence_count: integer, total number of sentences
015.       :param word_count: integer, total number of words
016.       :param syllable_count: integer, total number of syllables
017.       :return: integer, Flesh Kincaid grade level
018.       '''
019.            return  0.39  *  (word_count  /  sentence_count)  +  11.8  *
   (syllable_count / word_count) - 15.59
```

На вхід приймаються параметри, отримані від обробки таблиць заданих користувачем текстів.

На вихід програмний код видає розраховані індекси, для яких було прийняте рішення зробити власну інтерпретацію, що приміряна до рівнів CEFR. Її подано в Табл. 4.

Табл. 4: Інтерпретація індексів читабельності в рамках даного дослідження

Значення індексу ARI	***Значення індексу рівня Флеша-Кінкейда***	***Рівень CEFR***
[1; 2]	[0; 3)	A1
[3; 4]	[3; 6)	A2
[5; 6; 7]	[6; 9)	B1
[8; 9; 10]	[9; 12)	B2
[11; 12]	[12; 15)	C1
[13; 14]	[15;18]	C2

Також для Рівнів CEFR, які є порядковим типом даних (тобто нечислові дані, які мають певний порядок, і. е. A1<A2<B1 і так далі), було вирішено зробити числове перетворення у шкалу цілих чисел 1-6 для виконання простих

статистичних операцій, таких як знаходження максимального значення у вибірці, або середнього арифметичного.

Задля інтерпретації значень індексів було написано код, що перетворює числові значення індексів на рівні згідно Табл. 4:

```
001.  def interpret_ARI(index):
002.      rounded_index = round(index)
003.      if rounded_index == 2 or rounded_index == 1:
004.          return 'A1'
005.      elif rounded_index == 3 or rounded_index == 4:
006.          return 'A2'
007.       elif rounded_index == 5 or rounded_index == 6 or rounded_index ==
   7:
008.          return 'B1'
009.       elif rounded_index == 8 or rounded_index == 9 or rounded_index ==
   10:
010.          return 'B2'
011.      elif rounded_index == 11 or rounded_index == 12:
012.          return 'C1'
013.      elif rounded_index == 13 or rounded_index == 14:
014.          return 'C2'
015.      else:
016.          return '?'
017.
018.  def interpret_Flesh_Kincaid_grade(index):
019.      if index < 3:
020.          return 'A1'
021.      elif index < 6:
022.          return 'A2'
023.      elif index < 9:
024.          return 'B1'
025.      elif index < 12:
026.          return 'B2'
027.      elif index < 15:
028.          return 'C1'
```

```
029.      else:
030.          return 'C2'
```

2.2.2 Індекс граматичних компетенцій УМІ

Для того, щоб програма могла враховувати граматичні компетенції стандарту УМІ, було створено власний індекс граматичних компетенцій УМІ.

Для цього індекса кожному токену, грамемам кожного токена, що були розмічені pymorphy2 в цьому випадку, присвоюється окремий рівень CEFR згідно каталогу B в стандарті [17] та приблизним морфологічним тегам, що позначають деякі морфологічні та синтаксичні компетентності.

Повну таблицю переходу між грамемами та рівнями поміщено у Додаток 5

Приблизне прирівнювання морфологічних тегів до граматичних вимог стандарту було однією з поставлених задач дослідження. Було створено словник окремо взятих грамем, поділених за рівнями. Найбільший успіх станом на зараз саме у ідентифікації морфологічних показників; синтаксичні та словотвірні параметри покриті в деяких випадках або частково, або ніяк.

Суттєвою проблемою словника було те, що він містив дублікати тегів для різних рівнів. У зв’язку з тим, що для потреб програми необхідно, щоб кожен граматичний тег співвідносився лише з одним рівнем CEFR, було прийняте рішення модифікувати словник, що складається з підмножин для кожного рівня наступним чином:

$$A2 = A2 - A1 \cap A2;$$
$$B1 = B1 - A2 \cap B1;$$
$$B2 = B2 - B1 \cap B2;$$
$$C1 = C1 - B2 \cap C1;$$
$$C2 = C2 - C1 \cap C2;$$

Ця модифікація має допустити лише унікальні для кожного рівня теги, властиві лише їм, оскільки $A1 \in A2 \in B1 \in B2 \in C1 \in C2$. Також дані модифіковані списки передбачають, що програма матиме змогу ідентифікувати грамему слова і спочатку призначити її найнижчому можливому рівню, де цей тег знаходиться.

Втім, кількість тегів у словнику, у зв'язку з даною трансформацією, суттєво зменшиться.

Також постала проблема з грамемами та їхніми комбінаціями, що не мають визначеного рівня CEFR, або ж визначити їх лише за формальними ознаками чи механічним чином не є можливим з використанням тегів бібліотеки pymorphy2.

Також для ідентифікації рівня окремих токенів було взято їхні леми, а не грамеми, для ідентифікації. Наприклад, вимога складених форм порівняння прикметників та прислівників (що мають вивчатися на рівні A2) програмою ідентифікується за наявністю в тексті лем *більш, менш, більше, менше, найбільш, найменш, найбільше, найменше*[2]. Втім, часом виникала потреба додавати леми слів, що не мають відношення до вимог стандарту, суто з технічних причин помилкового розпізнавання їх як лем словоформ бібліотекою pymorphy2. Так, наприклад, у вимогу для B1 «утворення форм наказового способу» було додано окрім лем *хай, нехай* також лему *нехаяти* у зв'язку з тим, що pymorphy2 у багатьох випадках приписує саме цю лему до випадків, коли зустрічає токен *нехай* у тексті.

Втім, згодом ідею відкидання тегів на перетинів рівнів було скасовано, адже тоді тегів на деякі рівні (C2) не залишається. Натомість було запропоновано інший підхід:

---

2 У даному випадку бібліотека pymorphy2 розпізнає ці форми як окремі леми в деяких випадках, що може суперечити лінгвістичному глузду, але з технічних причин у список лем було додано саме такий набір.

1. Програма перевіряє слово на всі рівні CEFR
2. Програма нараховує слову кожен рівень, де зустрівся морфологічний тег або лема з рівневого словника
3. Програма обирає найнижчий рівень, де цей тег/лема знаходиться
4. Токену присвоюється рівень CEFR

У фінальному проєкті рівневий словник лем та грамем виглядає наступним чином:

```
001.  grammemes_dict = {
002.         'A1': [['NOUN', 'sing'], ['NOUN', 'plur', 'nomn'], ['NOUN',
   'accs', 'sing'], ['NOUN', 'loct', 'sing'], ['NOUN', 'voct'], ['ADJF',
   'sing'], ['ADJF', 'plur'], ['ADJF', 'nomn'], ['ADJF', 'accs'], ['ADJF',
   'loct'], ['NPRO', 'pers'], ['NPRO'], ['VERB', 'plur'], ['VERB', 'perf',
   'futr'], ['VERB', 'impr']],
003.         'A2': [['NOUN'], ['ADJF', 'sing'], ['ADJF', 'plur'], ['NPRO',
   'pers'], ['NPRO'], ['VERB', 'pres'], ['VERB', 'pres'], ['VERB', 'past'],
   ['VERB', 'futr'], ['ADJF', 'COMP'], ['ADJF', 'Supr'], ['ADJF', 'COMP'],
   ['ADJF', 'Supr']],
004.        'B1': [['NOUN'], ['ADJF'], ['NUMR'], ['NPRO'], ['VERB', 'Refl',
   'pres'], ['VERB', 'pres'], ['VERB', 'past', 'impf'], ['VERB', 'past',
   'perf'], ['VERB', 'futr', 'impf'], ['VERB', 'futr', 'perf'], ['VERB'
   'impr'], ['ADJF' 'COMP'], ['ADJF' 'Supr'], ['ADVB' 'COMP'], ['ADVB'
   'Supr']],
005.         'B2': [['NOUN'], ['NOUN' 'Abbr'], ['ADJF'], ['VERB' 'impf',
   'pres'], ['VERB' 'impf', 'futr'], ['VERB', 'pres'], ['VERB', 'futr'],
   ['VERB' 'perf'], ['VERB' 'impr'], ['ADJF' 'COMP'], ['ADJF' 'Supr'],
   ['ADVB' 'COMP'], ['ADVB' 'Supr']],
006.         'C1': [['NOUN'], ['NOUN' 'Abbr'], ['ADJF'], ['ADJF' 'COMP'],
   ['ADJF' 'Supr']],
007.       'C2': [['NOUN'], ['ADJF'], ['NPRO']]
008.  }
009.
010.  lemmas_dict = {
```

```
011.      'A2': ['боротися', 'більш', 'менш', 'більше', 'менше', 'найбільш',
   'найменш', 'найбільше', 'найменше'],
012.          'B1': ['себе', 'соб', 'той', 'цей', 'один', 'два', 'три',
   'чотири',    'багато', 'кілька', 'декілька', 'більш', 'менш', 'більше',
   'менше', 'найбільш', 'найменш', 'найбільше', 'найменше'],
013.        'B2': ['їхній', 'себе', 'соб', 'дати', 'їсти', 'хай', 'нехай',
   'нехаяти', 'би', 'б', 'більш', 'менш', 'більше', 'менше', 'найбільш',
   'найменш', 'найбільше', 'найменше', 'якщо', 'хоч', 'якби'],
014.          'C1': ['себе', 'соб', 'більш', 'менш', 'більше', 'менше',
   'найбільш', 'найменш', 'найбільше', 'найменше'],
015.      'C2': ['себе', 'соб']
016.  }
```

2.3 Огляд результатів роботи програми

В результаті роботи було написано програму для аналізу тексту за рівнями. Ця програма має графічний інтерфейс користувача для аналізу текстових файлів на основі їх рівня складності і розуміння з використанням індексів читабельності та граматичних рівнів. Програма розроблена на мові програмування Python із використанням бібліотеки PyQt6 для створення графічного інтерфейсу.

Інтерфейс програми збережено у файлі `main.py` (Додаток 1) При запуску програми користувач має надати адресу розташування txt-файлу, що аналізуватиметься. Скріншот інтерфейсу подано на Рис. 1.

Програма містить головний клас `window`, що успадковується від `QMainWindow`. Клас відповідає за створення і керування графічним інтерфейсом користувача, включаючи такі елементи як кнопки, мітки, і діалоги.

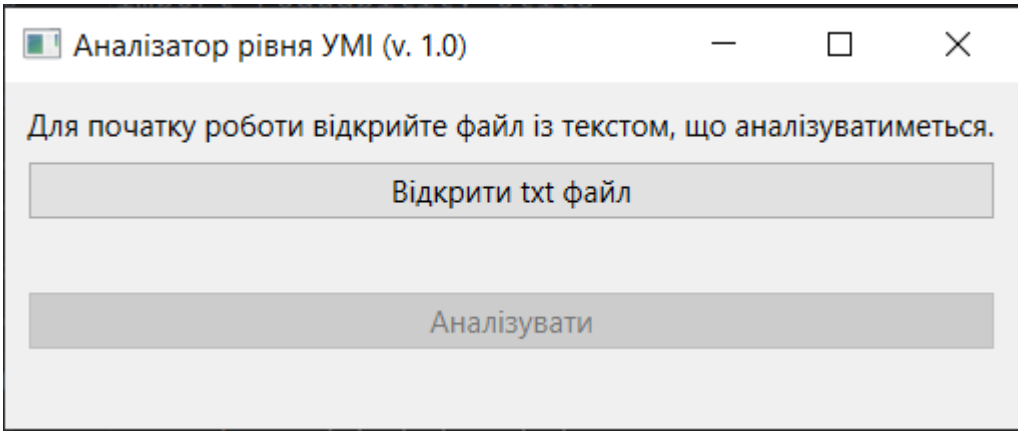

Рис. 1: Скріншот інтерфейсу програми при запуску

Основними елементами інтерфейсу є:

1. Кнопка відкриття файлу (open_button): відкриває діалогове вікно для вибору текстового файлу для аналізу.
2. Мітка для відображення інструкцій (open_label): інформує користувача про необхідність вибору файлу.
3. Мітка для відображення імені файлу (filename_label): показує шлях до обраного файлу.
4. Кнопка аналізу (analyze_button): запускає процес аналізу тексту, активується після вибору файлу.
5. Мітка для відображення результатів аналізу (analysis_label): показує результати аналізу тексту.
6. Кнопка збереження звіту (save_report_button): відкриває діалог для збереження текстового звіту.
7. Кнопка збереження таблиці слів (save_word_table_button): відкриває діалог для збереження таблиці слів у форматі CSV.
8. Кнопка збереження таблиці речень (save_sentences_table_button):

відкриває діалог для збереження таблиці речень у форматі CSV.

Процес роботи користувача з програмою виглядає наступним чином:

1. Відкриття файлу: Користувач вибирає текстовий файл для аналізу. Вміст файлу зчитується і зберігається в змінній analyzed_text. Після цього активується кнопка аналізу.
2. Аналіз тексту: При натисканні на кнопку аналізу, виконується наступне:
    1. Створюються таблиці слів (word_table) та речень (sent_table) з використанням функцій з модуля data_utils.
    2. Обчислюються індекси читабельності ARI та Flesch-Kincaid з використанням функцій з модуля readability_utils.
    3. Перевіряються грамемні рівні слів за допомогою модуля cefr_check_utils.
    4. Формується текстовий звіт про аналіз, який включає кількість речень і токенів у тексті, рівні читабельності та грамемні рівні.
3. Відображення результатів: Результати аналізу відображаються у відповідній мітці. Активуються кнопки для збереження звіту, таблиць слів і речень (Рис. 3).

Користувач має можливість зберегти результати аналізу у вигляді текстового звіту або таблиць у форматі CSV. Це здійснюється за допомогою діалогових вікон, які відкриваються при натисканні відповідних кнопок (див. Рис. 3).

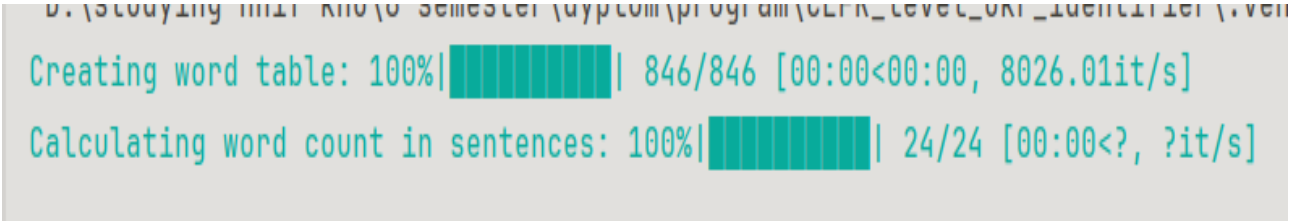

Рис. 2: Консольний рядок завантаження

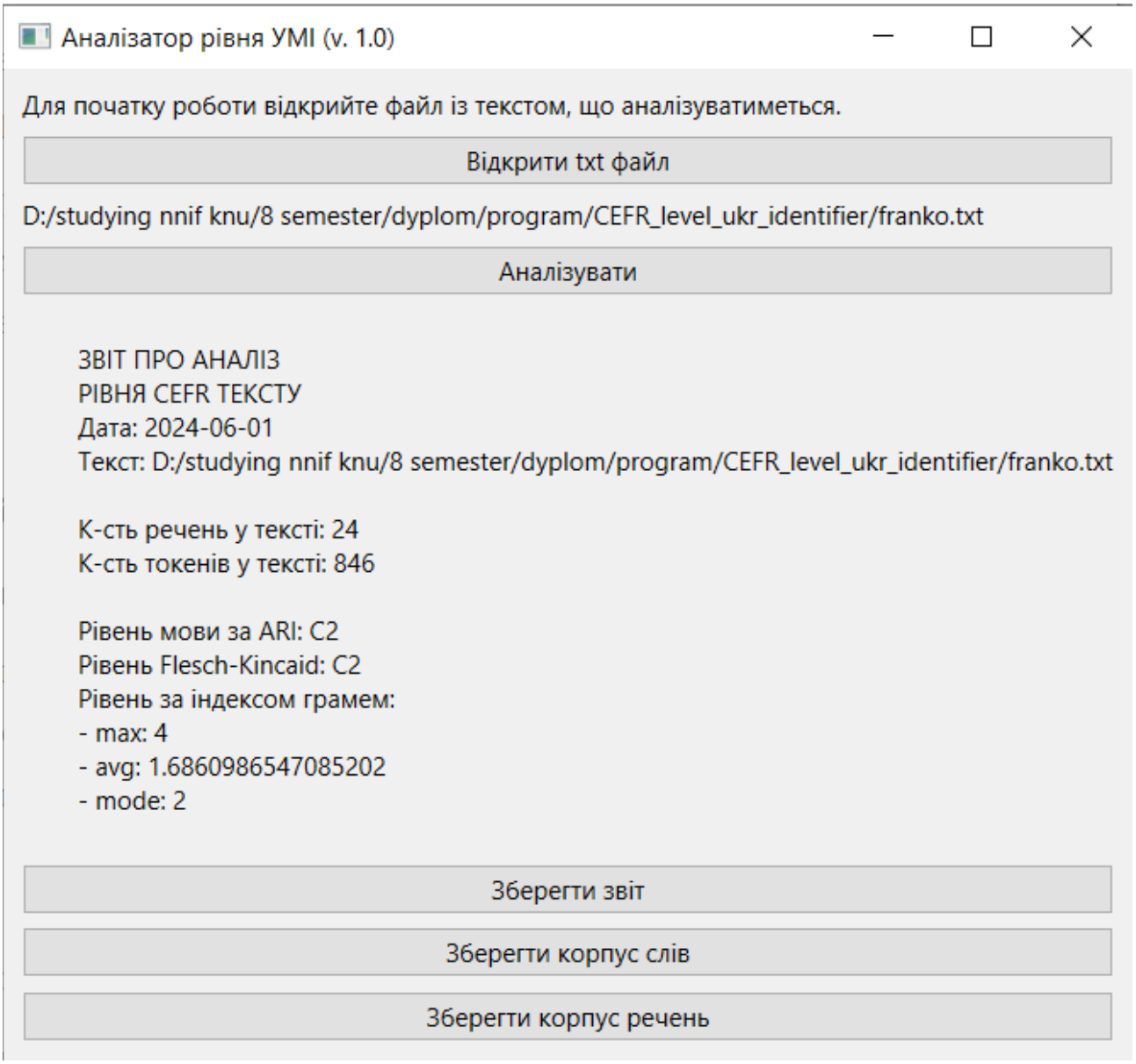

Рис. 3: Скріншот роботи програми після натискання кнопки "Аналізувати"

Ця програма є інструментом для аналізу текстів на рівень складності і

розуміння, надаючи користувачу детальний звіт про різні індекси читабельності та граматичні характеристики тексту. Завдяки зручному графічному інтерфейсу, користувач може легко аналізувати тексти та зберігати результати у потрібному форматі.

Програма «Аналізатор рівня УМІ» надає можливість збереження результатів аналізу у вигляді текстового звіту та табличних даних (таблиць слів і речень (Рис. 4, Рис. 5)). Нижче наведено детальний опис процесу збереження цих даних.

Збереження текстового звіту виконується за допомогою функції `save_report_dialog`. Користувач натискає кнопку «Зберегти звіт», після чого відкривається діалогове вікно для вибору місця збереження та імені файлу. Файли зберігаються у форматі .txt. Якщо користувач вибрав місце збереження, програма відкриває файл для запису і записує в нього текст звіту, який зберігається у змінній report. Мітка `save_label` оновлюється, щоб показати шлях до збереженого файлу, і стає видимою.

Також у консолі під час аналізу тексту виведено приблизний графічний рядок очікування завантаження аналізу слів та речень, написаний за допомогою бібліотеки tqdm [26].

11	Жара	жара	NOUN,femn,inan nomn	4	2	2
12	стояла	стояти	VERB,impf femn,past	6	3	2
13	страшенна	страшенний	ADJF femn,nomn	9	3	1
14	.	.	PNCT	1	0	0
15	Крізь	крізь	PREP,rv_accs	5	1	0
16	вітворені	вітворений	ADJF,pssv,impf plur,nomn	9	4	1
17	вікна	вікно	NOUN,neut,inan gent	5	2	2
18	вагона	вагон	NOUN,inan masc,gent	6	3	2
19	так	так	PRCL	3	1	0
20	і	і	CONJ,coord	1	1	0
21	сипався	сипатися	VERB,Refl,impf masc,past	7	3	2
22	вугляний	вугляний	ADJF masc,nomn	8	3	1

Рис. 4: Приклад сформованої таблиці-корпусу токенів

	sents	word_count
0	Літом 1895 року проїжджав я	11
1	Жара стояла страшенна.	4
2	Крізь вітворені вікна вагона т	15
3	Ми поспускали стори в купе і	33
4	Не дуже далеко від Будапеш	20
5	Один був старший пан, висо	47
6	Ті самі прикмети чути було і	24
7	З усеї зверхньої подоби, з од	33
8	Його товариш був молодий,	41
9	І молодий хлопець, хоть у нь	35

Рис. 5: Приклад сформованої таблиці-корпусу речень

2.4 Аналіз результатів роботи програми

Для аналізу роботи моделі у неї було введено кілька перших абзаців з наступних текстів:

1. Художній стиль

1. О. Забужко “Сестро, сестро... ”[3] (Довжина уривка 277 токенів)
2. В. Підмогильний “Місто”[4] (Довжина уривка 226 токенів)
3. С. Жадан “Пливи, рибо, пливи... ”[5] (Довжина уривка 191 токен)

2. Офіційно-діловий стиль
    1. ЗУ “Про вищу освіту”[6] (Довжина уривка 201 токен)
    2. ЗУ “Про освіту”[7] (Довжина уривка 291 токен)
    3. Конституція України[8] (Довжина уривка 281 токен)

Після обробки програма видала результати рівнів за метриками (Табл. 5).

Показники середнього значення та моди граматичних індексів було виключено з аналізу через їх обмежений технічний характер. Ці показники надають лише інформацію про те, які граматичні конструкції найчастіше зустрічаються в тексті, без врахування найвищого присутнього аспекту мовної складності.

Табл. 5: Результати обрахунків індексів програмою

Назва твору, з якого взято уривок	***Індекс ARI***	***Індекс Флеша-Кінкейда***	***Граматичний індекс (максимум)***
Сестро, сестро...	C2	C2	4 (B2)
Місто	C1	C1	3 (B1)

3 https://www.ukrlib.com.ua/books/printit.php?tid=4203

4 https://www.ukrlib.com.ua/books/printit.php?tid=76

5 https://www.ukrlib.com.ua/books/printit.php?tid=18088

6 https://zakon.rada.gov.ua/go/1556-18

7 https://zakon.rada.gov.ua/go/2145-19

8 https://zakon.rada.gov.ua/go/254%D0%BA/96-%D0%B2%D1%80

Назва твору, з якого взято уривок	***Індекс ARI***	***Індекс Флеша-Кінкейда***	***Граматичний індекс (максимум)***
Пливи, рибо, пливи...	A2	B1	4 (B2)
Про вищу освіту	C2	C2	3 (B1)
Про освіту	C2	C2	3 (B1)
Конституція України	C2	C2	3 (B1)

Табл. 5 вказує на помітні розбіжності між рівнями, обчисленими за допомогою граматичних показників, та рівнями, визначеними за індексами читабельності. Така невідповідність може бути зумовлена обмеженнями граматичних індексів у відображенні позаграматичних аспектів мови. Наприклад, текст може виявитися менш придатним для читання для користувача з рівнем володіння мовою A2, навіть якщо його граматичні характеристики відповідають рівню A1. Це свідчить про те, що граматичні індекси не завжди можуть адекватно оцінити складність тексту з погляду читабельності.

Також Табл. 5 демонструє, що граматичні індекси можуть мати обмежену здатність присвоювати текстам рівні, вищі за B2. Це явище можна пояснити загальністю граматичних вимог, які пред'являються до рівнів C1 та C2. Відповідно до рекомендацій, на цих рівнях вважається, що мовний користувач має володіти практично всіма граматичними структурами. Таким чином, існуючі граматичні індекси можуть не враховувати специфічні особливості, які

відрізняють тексти на рівнях C1 і C2, що ускладнює їх точну оцінку.

Крім того, існує потреба в більш детальному вивченні того, як індекси читабельності враховують лексичне багатство, синтаксичну складність та прагматичні особливості тексту. Наприклад, лексичне багатство може включати варіативність словникового запасу, наявність рідковживаних слів та ідіоматичних виразів, які можуть значно вплинути на читабельність та рівень УМІ. Синтаксична складність, у свою чергу, враховувала б кількість підрядних речень, використання пасивних конструкцій та інших складних синтаксичних структур.

Ці спостереження вказують на необхідність подальшого вдосконалення методів оцінювання мовної складності текстів, зокрема, шляхом інтеграції позаграматичних параметрів, таких як семантична складність, структурна організація тексту та комунікативні наміри, описані в стандарті. Такий підхід дозволить забезпечити більш комплексну та точну оцінку текстів, що враховує не лише граматичну правильність, але й інші важливі аспекти, що впливають на читабельність та зрозумілість тексту для різних категорій користувачів.

2.5 Обмеження дослідження

Розроблена програма має певні обмеження у своїй здатності оцінювати рівень володіння мовою згідно з загальноєвропейськими рекомендаціями з мовної освіти (CEFR) для української мови як іноземної (УМІ) [2; 17]. По-перше, запропонований алгоритм не враховує змістовні аспекти мовлення, такі як вимоги до тематик, стилів та комунікативних ролей. Програмний код здатний розпізнати, наприклад, що текст містить вживання питального займенника «скільки», що відповідає компетенції рівня B2, але не може розпізнати комунікативний намір вираження протесту, що є компетенцією

рівня C2 [17]. Це обмеження демонструє, що хоча алгоритм може обробляти окремі граматичні елементи, він не здатний інтерпретувати більш складні комунікативні контексти, що є важливим для вищих рівнів мовної компетенції.

Крім того, оптимізованість індексу граматичних компетенцій для УМІ може призводити до помилкових результатів через недостатньо деталізовану морфологічну розмітку. Наприклад, програма може розпізнати відмінювання іменників, що відповідає компетенціям вищих рівнів, але не здатна врахувати нюанси. У стандарті зазначено відмінювання різних форм іменників (м'яка, тверда групи) на різних рівнях володіння мовою [17]. Це означає, що програма може ідентифікувати базові морфологічні форми, але не враховує тонкі відмінності у використанні цих форм на різних рівнях мовної компетенції.

Завдання додавання цих нюансів до програмного коду або розмітки є складним і наразі неможливе з використанням наявної анотації та інструментів. Однак, ця проблема може бути розглянута як потенціал для майбутніх досліджень у контексті створення спеціалізованих інструментів для розмітки морфологічних та словотвірних параметрів української мови. Подальші дослідження можуть спрямовуватися на розробку більш детальних та точних анотацій, що враховуватимуть специфічні мовні явища, та на удосконалення алгоритмів для більш глибокого аналізу текстів, що дозволить підвищити точність оцінки мовної компетенції відповідно до стандартів CEFR.

У підсумку, хоч ця програма і здатна автоматично визначати певні граматичні структури, необхідні для оцінки рівня мовної компетенції, її обмеження у сфері аналізу змістових планів мовлення та морфологічної розмітки підкреслюють важливість подальших досліджень і розробок у цій галузі. Це сприятиме створенню більш точних та універсальних інструментів для оцінки мовної компетенції, що відповідають сучасним вимогам та

стандартам мовної освіти.

Також варто звернути особливу увагу на використання індексів читабельності у даному дослідженні. Зокрема, індекси ARI (Automated Readability Index) та Flesch-Kincaid Readability Tests спочатку були розроблені для аналізу англомовних текстів. Формули, на яких базуються ці індекси, були створені шляхом застосування методів лінійної регресії, що враховували специфіку англійської мови [9]. Це означає, що вони були адаптовані до структурних та лексичних особливостей англійської мови, таких як середня довжина слів і речень, частота використання складних слів тощо.

Використання цих індексів без адаптації до україномовних текстів призводить до певних невідповідностей та може спотворювати результати оцінки читабельності. Українська мова має свої власні лінгвістичні характеристики, які відрізняються від англійської. Наприклад, українські слова можуть бути довшими за кількістю літер через велику кількість суфіксів і префіксів, а синтаксична структура речень може суттєво відрізнятися від англійської. Такі відмінності впливають на показники читабельності, якщо застосовувати до них формули, розроблені для іншої мовної системи.

З іншого боку, постає питання, які інструменти використовувати для оцінки читабельності текстів українською мовою, оскільки наразі комп'ютерними лінгвістами не було розроблено відповідних адаптованих індексів. Відсутність спеціалізованих індексів для української мови ускладнює проведення досліджень у цій галузі та може обмежувати можливості аналізу текстів.

Ця проблема відкриває перспективи для подальших досліджень і розробок у сфері комп'ютерної лінгвістики. Зокрема, необхідно розробити нові індекси читабельності, які будуть враховувати специфіку української мови. Це може

включати створення нових формул на основі корпусних досліджень українських текстів, які відображатимуть середню довжину слів і речень, частоту використання різних граматичних структур та інших лінгвістичних характеристик, властивих українській мові.

У підсумку, хоча індекси ARI та Flesch-Kincaid є корисними інструментами для оцінки читабельності текстів англійською мовою, їх використання для української мови без належної адаптації не є оптимальним. Це підкреслює важливість подальших досліджень, спрямованих на розробку спеціалізованих інструментів для аналізу українських текстів, що дозволить забезпечити точність і релевантність оцінки читабельності для різних мовних контекстів.

Висновки

У результаті проведених досліджень було розроблено програму для аналізу тексту, яка здійснює граматичну анотацію, визначає параметри тексту (такі як довжина слова у символах та складах) і надає звіт щодо ймовірного рівня володіння мовою згідно CEFR. Крім того, програма генерує таблиці з проанотованими токенами та реченнями.

Проте, під час розробки виникла низка проблем. Однією з них є відсутність «золотого стандарту» для української мови, тобто корпусу текстів, проанотованих за рівнями CEFR. Ця проблема ускладнює точне визначення рівня мовного володіння. Створення такого корпусу могло б вирішити цю проблему та покращити результати програми.

Крім того, важливим є застосування моделей машинного навчання для підвищення точності визначення рівня тексту, що неможливе без наявності вищезгаданого «золотого стандарту». Також, розробка спеціальних словників лексем за тематичними каталогами рівнів CEFR може сприяти поліпшенню моделей ідентифікації рівня тексту, що допоможе виявити комунікативний зміст тексту.

У разі наявності «золотого стандарту» ми зможемо розробити власні формули читабельності для українськомовних текстів з використанням лінійної регресії, які враховуватимуть специфіку української мови і параметри, характерні саме для неї. Це дозволить отримати більш точні та адаптовані до української мови формули, що відрізнятимуться від тих, які використовуються для англійської мови. Такий підхід забезпечить більш точне визначення рівня складності текстів українською мовою і поліпшить якість аналізу текстів для використання в освітніх та дослідницьких цілях.

У перспективі, за наявності так званого «золотого стандарту» ми можемо розробити власні індекси читабельності, використовуючи методи лінійної регресії, та вивести унікальні формули для їх обчислення. Ці формули можна буде інтегрувати у власні нейронні моделі класифікації тексту як додаткові параметри. Такий підхід дозволить підвищити точність та ефективність класифікаційних алгоритмів, що використовуються для оцінки текстів у контексті різноманітних лінгвістичних задач.

Більш того, створення таких індексів та їх інтеграція в нейронні моделі може мати значний вплив на автоматизацію процесів у лінгводидактиці та тестології. Автоматизовані системи, збагачені новими параметрами читабельності, зможуть надавати більш точні й комплексні оцінки текстів, що важливо для розробки навчальних матеріалів, тестових завдань та інших освітніх ресурсів. Це, у свою чергу, сприятиме підвищенню якості мовної освіти, дозволяючи викладачам і дослідникам ефективніше аналізувати та адаптувати тексти до потреб різних категорій учнів.

Крім того, такі розробки можуть стати каталізатором для подальших інновацій у сфері мовної освіти, відкриваючи нові можливості для створення адаптивних навчальних програм, які автоматично підлаштовуються під рівень підготовки та індивідуальні особливості учнів. Використання новітніх технологій у комбінації з глибокими лінгвістичними знаннями дозволить розробляти більш досконалі та ефективні методи навчання, що відповідатимуть сучасним вимогам та стандартам освіти.

У підсумку, можна відзначити, що програма, хоч і потребує подальших досліджень та вдосконалень, вже сьогодні є значним кроком у напрямку автоматизації визначення рівня мовної компетенції. З урахуванням можливих перспектив, її можна розглядати як потенційно цінний інструмент для вчителів

української мови. У майбутньому вона може стати важливою підтримкою у процесі аналізу та оцінки текстів з метою підвищення ефективності навчання та розвитку мовної компетенції учнів.

Перелік джерел посилання

1. Ясуда К.-К. Створення програми для автоматичного визначення рівня володіння мовою за шкалою CEFR тексту українською мовою : Курсова робота. Київ : Київський нацональний університет імені Тараса Шевченка, 2023.

2. За ред. Europarat. Common European framework of reference for languages: learning, teaching, assessment. 10th print. Cambridge : Cambridge Univ. Press, 2010. 260 с. ISBN 978-0-521-80313-7.

3. Crystal D. The Cambridge encyclopedia of language. 3. ed. Cambridge : Cambridge University Press, 2010. 516 с. ISBN 978-0-521-73650-3.

4. datetime — Basic date and time types. *Python documentation*. URL: https://docs.python.org/3/library/datetime.html (дата звернення: 01.06.2024).

5. Flesch Reading Ease and the Flesch Kincaid Grade Level. *Readable*. URL: https://readable.com/readability/flesch-reading-ease-flesch-kincaid-grade-level/ (дата звернення: 10.06.2024).

6. Gaillat T., Simpkin A., Ballier N. та ін. Predicting CEFR levels in learners of English: The use of microsystem criterial features in a machine learning approach. *ReCALL*. Вип. 34, № 2. С. 130–146. DOI:10.1017/S095834402100029X.

7. Geertzen J., Alexopoulou T., Korhonen A. Automatic Linguistic Annotation of Large Scale L2 Databases: The EF-Cambridge Open Language Database (EFCamDat). 2014.

8. Ke Z., Ng V. Automated Essay Scoring: A Survey of the State of the Art. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial IntelligenceTwenty-Eighth International Joint Conference on Artificial Intelligence {IJCAI-19}*. Macao, China : International Joint Conferences on Artificial Intelligence Organization, 2019. DOI:10.24963/ijcai.2019/879. С. 6300–6308.

9. Kincaid J. P., Fishburne Jr., Robert P. R. та ін. Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel: (Fort Belvoir, VA, 01.02.1975). Fort Belvoir, VA : Defense Technical Information Center, 1975. DOI:10.21236/ADA006655. 1975.

10. Kincaid J. P., Braby R., Mears J. E. Electronic authoring and delivery of technical

information. *Journal of Instructional Development*. Вип. 11, № 2. С. 8–13. DOI:10.1007/BF02904998.

11. Korobov M. Morphological Analyzer and Generator for Russian and Ukrainian Languages. *Analysis of Images, Social Networks and Texts*. ред. Mikhail Yu. Khachay, Natalia Konstantinova, Alexander Panchenko, Dmitry I. Ignatov, Valeri G. Labunets. Springer International Publishing, 2015. С. 320–332. DOI:10.1007/978-3-319-26123-2_31.

12. The Python IDE for data science and web development with intelligent code completion, on-the-fly error checking, quick-fixes, and much more. PyCharm: the Python IDE for data science and web development. *JetBrains*. The Python IDE for data science and web development with intelligent code completion, on-the-fly error checking, quick-fixes, and much more. URL: https://www.jetbrains.com/pycharm/ (дата звернення: 26.05.2024).

13. Santucci V., Santarelli F., Forti L. та ін. Automatic Classification of Text Complexity. *Applied Sciences*. Вип. 10, № 20. С. 7285. DOI:10.3390/app10207285.

14. Smith E. A., Senter R. J. Automated readability index. *AMRL-TR. Aerospace Medical Research Laboratories (U.S.)*. 05.1967. С. 1–14.

15. Антонів О., Бойко Г., Синчак О. Українська мова як іноземна. Рівні загального володіння A1 – C2. (Київ, 2024). Київ, 2024Also available online, URL: https://mova.gov.ua/gromadskosti/rehuliatorna-diialnist-ta-konsultatsii-z-hromadskistiu/obgovorennya-proyektiv-dokumentiv/hromadski-obhovorennia-2024/proiektu-standartu-derzhavnoi-movy-ukrainska-mova-iak-inozemna-rivni-zahalnoho-volodinnia-a1-s2 2024.

16. Левчук П. Як національна комісія українську мову в ЄС проводжала. *LB.ua*. 31.05.2024. URL: https://lb.ua/blog/pavlo_levchuk/616287_yak_natsionalna_komisiya_ukrainsku.html (accessed 02/06/2024).

17. Мазурик Д., Антонів О., Синчак О. et al. Стандартизовані вимоги до рівнів володіння українською мовою як іноземною A1-C2. (Київ, 2018). Київ : Міністерство освіти і науки України, 2018. 2018.

18. Ясуда К.-К., Костіков М. Проєктування програмного засобу для ідентифікації рівня володіння українською мовою як іноземною. *Матеріали89 Міжнародна наукова конференція молодих учених, аспірантів і студентів «Наукові здобутки молоді — вирішенню проблем харчування людства у XXI столітті»*.

Vol. 2, 07.07.2023. (Київ, 07.07.2023). Київ : НУХТ, 2023. P. 345.

19. Elma Kerz, Yu Qiao, Daniel Wiechmann et al. Automated Classification of Written Proficiency Levels on the CEFR-Scale through Complexity Contours and RNNs. *16th Workshop on Innovative Use of NLP for Building Educational Applications*. P. 199–209.

20. François T., Fairon C. An "AI readability" Formula for French as a Foreign Language. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*(Jeju Island, Korea, 07.2012). Jeju Island, Korea : Association for Computational Linguistics, 2012. Also available online, URL: https://aclanthology.org/D12-1043 P. 466–477.

21. Limited R. C. PyQt6: Python bindings for the Qt cross platform application toolkit.

22. Marcus S., Kerz E., Wiechmann D. et al. CoCoGen - Complexity Contour Generator: Automatic Assessment of Linguistic Complexity Using a Sliding-Window Technique. *Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity (CL4LC)CL4LC 2016*. Osaka, Japan : The COLING 2016 Organizing Committee, 2016. Also available online, URL: https://aclanthology.org/W16-4103 (accessed 09/05/2023).P. 23–31.

23. McKinney W. Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference*(2010). DOI:10.25080/Majora-92bf1922-00a. P. 56–61.

24. Python 3.8.0 Documentation. URL: https://docs.python.org/release/3.8.0/ (accessed 09/04/2023).

25. team T. pandas development. pandas-dev/pandas: Pandas. (02.2020). Zenodo, 2020. DOI:10.5281/zenodo.3509134. 2020.

26. tqdm documentation. URL: https://tqdm.github.io/ (accessed 01/06/2024).

27. Sai A. B., Mohankumar A. K., Khapra M. M. A Survey of Evaluation Metrics Used for NLG Systems. arXiv, 2020. DOI:10.48550/ARXIV.2008.12009.

Додаток 1. Програмний код файлу «main.py»

```
001)  import cefr_check_utils
002)  import data_utils
003)  import readability_utils
004)  import pandas as pd
005)
006)  from datetime import date
007)
008)  from PyQt6.QtWidgets import QApplication, QWidget, QMainWindow, QLabel, QPushButton, QVBoxLayout, QFileDialog
009)  import sys
010)
011)  import readability_utils
012)
013)
014)  class window(QMainWindow):
015)      def __init__(self):
016)          super().__init__()
017)
018)          self.setWindowTitle('Аналізатор рівня УМІ (v. 1.0)')
019)
020)          self.open_button = QPushButton(self)
021)          self.open_button.setText('Відкрити txt файл')
022)          self.open_button.clicked.connect(self.open_file_dialog)
023)
024)          self.open_label = QLabel(self)
025)              self.open_label.setText('Для початку роботи відкрийте файл із текстом, що аналізуватиметься.')
026)
027)          self.filename_label = QLabel(self)
028)
029)          self.analyze_button = QPushButton(self)
030)          self.analyze_button.setText('Аналізувати')
031)          self.analyze_button.setEnabled(False)
032)          self.analyze_button.clicked.connect(self.analyze_text)
033)
034)          self.analysis_label = QLabel(self)
```

```
035)
036)          self.save_report_button = QPushButton(self)
037)          self.save_report_button.setText('Зберегти звіт')
038)          self.save_report_button.setHidden(True)
039)          self.save_report_button.clicked.connect(self.save_report_dialog)
040)
041)          self.save_label = QLabel(self)
042)          self.save_label.setHidden(True)
043)
044)          self.save_word_table_button = QPushButton(self)
045)          self.save_word_table_button.setText('Зберегти корпус слів')
046)          self.save_word_table_button.setHidden(True)
047)          self.save_word_table_button.clicked.connect(self.save_word_table_d
ialog)
048)
049)          self.save_sentences_table_button = QPushButton(self)
050)          self.save_sentences_table_button.setText('Зберегти корпус речень')
051)          self.save_sentences_table_button.setHidden(True)
052)          self.save_sentences_table_button.clicked.connect(self.save_sentenc
e_table_dialog)
053)
054)          layout = QVBoxLayout()
055)          layout.addWidget(self.open_label)
056)          layout.addWidget(self.open_button)
057)          layout.addWidget(self.filename_label)
058)          layout.addWidget(self.analyze_button)
059)          layout.addWidget(self.analysis_label)
060)          layout.addWidget(self.save_report_button)
061)          layout.addWidget(self.save_label)
062)          layout.addWidget(self.save_word_table_button)
063)          layout.addWidget(self.save_sentences_table_button)
064)
065)          container = QWidget(self)
066)          container.setLayout(layout)
067)
068)          self.setCentralWidget(container)
069)
070)      def open_file_dialog(self):
```

```
071)          filename = QFileDialog.getOpenFileName(
072)              self,
073)              "Оберіть текстовий файл",
074)              "",
075)              "Text files (*.txt)"
076)          )
077)          if filename[0]:
078)              global analyzed_text
079)              with open(filename[0], 'r', encoding='utf-8') as f:
080)                  analyzed_text = f.read()
081)              self.filename_label.setText(filename[0])
082)              self.analyze_button.setEnabled(True)
083)
084)      def save_report_dialog(self):
085)          global report
086)          filename = QFileDialog.getSaveFileName(
087)              self,
088)              "Оберіть місце збереження звіту",
089)              "",
090)              "Text files (*.txt)"
091)          )
092)          if filename[0]:
093)              with open(filename[0], 'w', encoding='utf-8') as f:
094)                  f.write(report)
095)              self.save_label.setText(filename[0])
096)              self.save_label.setHidden(False)
097)
098)      def save_word_table_dialog(self):
099)          filename = QFileDialog.getSaveFileName(
100)              self,
101)              "Оберіть місце збереження корпусу слів",
102)              "",
103)              "Table files (*.csv)"
104)          )
105)          if filename[0]:
106)              word_table.to_csv(filename[0])
107)              self.save_label.setText(filename[0])
108)              self.save_label.setHidden(False)
```

```
109)
110)      def save_sentence_table_dialog(self):
111)          global sent_table
112)          filename = QFileDialog.getSaveFileName(
113)              self,
114)              "Оберіть місце збереження корпусу речень",
115)              "",
116)              "Table files (*.csv)"
117)          )
118)          if filename[0]:
119)              sent_table.to_csv(filename[0])
120)              self.save_label.setText(filename[0])
121)              self.save_label.setHidden(False)
122)
123)      def analyze_text(self):
124)          global analyzed_text, \
125)              word_table, \
126)              sent_table, \
127)              ARI, \
128)              Flesch_Kincaid, \
129)              grammeme_level_interpretation, \
130)              report
131)          word_table = data_utils.create_word_table(analyzed_text)
132)          sent_table = data_utils.create_sentence_table(analyzed_text)
133)          ARI = readability_utils.interpret_ARI(readability_utils.ARI_index(
134)              len(sent_table),
135)              len(word_table),
136)              word_table['word_len'].sum()
137)          ))
138)                                                      Flesch_Kincaid      =
readability_utils.interpret_Flesh_Kincaid_grade(readability_utils.Flesh_Kincaid_
grade_level(
139)              len(sent_table),
140)              len(word_table),
141)              word_table['syllable_count'].sum()
142)          ))
143)
144)          word_table = cefr_check_utils.check_grammemes_level(word_table)
```

```
145)                                          grammeme_level_interpretation     =
cefr_check_utils.interpret_cefr_list(word_table['cefr_level'])
146)
147)          report = f'''
148)          ЗВІТ ПРО АНАЛІЗ
149)          РІВНЯ CEFR ТЕКСТУ
150)          Дата: {date.today().strftime('%Y-%m-%d')}
151)          Текст: {self.filename_label.text()}
152)
153)          К-сть речень у тексті: {len(sent_table)}
154)          К-сть токенів у тексті: {len(word_table)}
155)
156)          Рівень мови за ARI: {ARI}
157)          Рівень Flesch-Kincaid: {Flesch_Kincaid}
158)          Рівень за індексом грамем:
159)          - max: {grammeme_level_interpretation['max']}
160)          - avg: {grammeme_level_interpretation['avg']}
161)          - mode: {grammeme_level_interpretation['mode']}
162)          '''
163)
164)          self.analysis_label.setText(report)
165)
166)          self.save_report_button.setHidden(False)
167)          self.save_word_table_button.setHidden(False)
168)          self.save_sentences_table_button.setHidden(False)
169)
170)
171)  if __name__ == '__main__':
172)      app = QApplication(sys.argv)
173)
174)      window = window()
175)
176)      window.show()
177)
178)      app.exec()
```

Додаток 2. Програмний код файлу «data_utils.py»

```
001)  import pandas as pd
002)  from tokenize_uk.tokenize_uk import tokenize_words, tokenize_sents
003)  import pymorphy2
004)  from tqdm import tqdm
005)
006)  morph = pymorphy2.MorphAnalyzer(lang='uk')
007)
008)
009)  def create_sentence_table(text):
010)      sents = tokenize_sents(text)
011)      # count words in sentence
012)      word_count_list = []
013)      for sent in tqdm(sents, desc='Calculating word count in sentences'):
014)          word_count_list.append(len(tokenize_words(sent)))
015)      return pd.DataFrame({'sents': sents, 'word_count': word_count_list})
016)
017)
018)  def count_ukr_syllables(text):
019)      """
020)      calculates the number of syllables (for ukrainian language) in a given
text string
021)      :param text: string object
022)      :return: syllable count (integer)
023)      """
024)      syllables = 0
025)      for char in text:
026)          if char in 'аеіоуиїяєю':
027)              syllables += 1
028)      return syllables
029)
030)
031)  def create_word_table(text):
032)      """
033)      creates a word table from a given text string
034)      :param text:
035)       :return: df containing all tokens from text and their linguistic text
```

```
parameters
036)      """
037)      tokens = tokenize_words(text)
038)      # get grammemes, lemmas, lengths, syllable count
039)      grammeme_list = []
040)      lemma_list = []
041)      word_len_list = []
042)      syllable_count_list = []
043)      for token in tqdm(tokens, desc='Creating word table'):
044)          grammeme_list.append(morph.parse(token)[0].tag)
045)          lemma_list.append(morph.parse(token)[0].normal_form)
046)          word_len_list.append(len(token))
047)          syllable_count_list.append(count_ukr_syllables(token))
048)
049)      return pd.DataFrame({'token': tokens, 'lemma': lemma_list, 'grammeme':
grammeme_list, 'word_len': word_len_list,
050)                           'syllable_count': syllable_count_list})
```

Додаток 3. Програмний код файлу «cefr_check_utils.py»

```
# dictionary of grammemes

grammemes_dict = {
      'A1': [['NOUN', 'sing'], ['NOUN', 'plur', 'nomn'], ['NOUN', 'accs', 'sing'], ['NOUN', 'loct', 'sing'],
             ['NOUN', 'voct'], ['ADJF', 'sing'], ['ADJF', 'plur'], ['ADJF', 'nomn'], ['ADJF', 'accs'], ['ADJF', 'loct'],
             ['NPRO', 'pers'], ['NPRO'], ['VERB', 'plur'], ['VERB', 'perf', 'futr'], ['VERB', 'impr']],
     'A2': [['NOUN'], ['ADJF', 'sing'], ['ADJF', 'plur'], ['NPRO', 'pers'], ['NPRO'], ['VERB', 'pres'], ['VERB', 'pres'],
             ['VERB', 'past'], ['VERB', 'futr'], ['ADJF', 'COMP'], ['ADJF', 'Supr'], ['ADJF', 'COMP'], ['ADJF', 'Supr']],
        'B1': [['NOUN'], ['ADJF'], ['NUMR'], ['NPRO'], ['VERB', 'Refl', 'pres'], ['VERB', 'pres'], ['VERB', 'past', 'impf'],
              ['VERB', 'past', 'perf'], ['VERB', 'futr', 'impf'], ['VERB', 'futr', 'perf'], ['VERB', 'impr'],
             ['ADJF', 'COMP'], ['ADJF', 'Supr'], ['ADVB', 'COMP'], ['ADVB', 'Supr']],
     'B2': [['NOUN'], ['NOUN', 'Abbr'], ['ADJF'], ['VERB', 'impf', 'pres'], ['VERB', 'impf', 'futr'], ['VERB', 'pres'],
             ['VERB', 'futr'], ['VERB', 'perf'], ['VERB', 'impr'], ['ADJF', 'COMP'], ['ADJF', 'Supr'], ['ADVB', 'COMP'],
            ['ADVB', 'Supr']],
        'C1': [['NOUN'], ['NOUN', 'Abbr'], ['ADJF'], ['ADJF', 'COMP'], ['ADJF', 'Supr']],
     'C2': [['NOUN'], ['ADJF'], ['NPRO']]
}

lemmas_dict = {
       'A2': ['боротися', 'більш', 'менш', 'більше', 'менше', 'найбільш', 'найменш', 'найбільше', 'найменше'],
      'B1': ['себе', 'соб', 'той', 'цей', 'один', 'два', 'три', 'чотири', 'багато', 'кілька', 'декілька', 'більш', 'менш',
                  'більше', 'менше', 'найбільш', 'найменш', 'найбільше', 'найменше'],
```

```
        'B2': ['їхній', 'себе', 'соб', 'дати', 'їсти', 'хай', 'нехай',
'нехаяти', 'би', 'б', 'більш', 'менш', 'більше',
                 'менше', 'найбільш', 'найменш', 'найбільше', 'найменше',
'якщо', 'хоч', 'якби'],
      'C1': ['себе', 'соб', 'більш', 'менш', 'більше', 'менше', 'найбільш',
'найменш', 'найбільше', 'найменше'],
     'C2': ['себе', 'соб']
 }


 # define the level of the grammemes
 def check_grammemes_level(df):
     """
     calculate cefr levels for each word in the word table.
     :param df: DataFrame, word table
     :return: DataFrame, word table with cefr_level column
     """
     level_list = []
     for index in df.index:
         temp_level_list = []
         # check grammemes level
         for grammeme in grammemes_dict['A1']:
             if set(grammeme) in df['grammeme'][index]:
                 temp_level_list.append(1)
         for grammeme in grammemes_dict['A2']:
             if set(grammeme) in df['grammeme'][index]:
                 temp_level_list.append(2)
         for grammeme in grammemes_dict['B1']:
             if set(grammeme) in df['grammeme'][index]:
                 temp_level_list.append(3)
         for grammeme in grammemes_dict['B2']:
             if set(grammeme) in df['grammeme'][index]:
                 temp_level_list.append(4)
         for grammeme in grammemes_dict['C1']:
             if set(grammeme) in df['grammeme'][index]:
                 temp_level_list.append(5)
         for grammeme in grammemes_dict['C2']:
             if set(grammeme) in df['grammeme'][index]:
```

```
058)                 temp_level_list.append(6)
059)         # check lemmas level
060)         for lemma in lemmas_dict['A2']:
061)             if df['lemma'][index] == lemma:
062)                 temp_level_list.append(2)
063)         for lemma in lemmas_dict['B1']:
064)             if df['lemma'][index] == lemma:
065)                 temp_level_list.append(3)
066)         for lemma in lemmas_dict['B2']:
067)             if df['lemma'][index] == lemma:
068)                 temp_level_list.append(4)
069)         for lemma in lemmas_dict['C1']:
070)             if df['lemma'][index] == lemma:
071)                 temp_level_list.append(5)
072)         for lemma in lemmas_dict['C2']:
073)             if df['lemma'][index] == lemma:
074)                 temp_level_list.append(6)
075)         # check if no levels were found
076)         if not temp_level_list:
077)             temp_level_list.append(0)
078)         # append token level
079)         level_list.append(min(temp_level_list))
080)     return df.assign(cefr_level=level_list)
081)
082)
083)  from statistics import mean, mode
084)
085)
086)  def interpret_cefr_list(level_list):
087)     """
088)      create a dictionary of max, average and mode interpretations of level
lists
089)     :param level_list: list of level indices converted to integer values
090)     :return: dictionary of max, average and mode interpretations
091)     """
092)     level_list_no_zeros = list(filter(lambda x: x != 0, level_list))
093)     if not level_list_no_zeros:
094)         level_list_no_zeros.append(1)
```

```
095)      interpretation_list = {
096)          'max': max(level_list),
097)          'mode': mode(level_list_no_zeros),
098)          'avg': mean(level_list_no_zeros)
099)      }
100)      return interpretation_list
```

Додаток 4. Програмний код файлу «readability_utils.py»

```
001.  def ARI_index(sent_count, word_count, char_count):
002.      '''
003.       Calculate ARI index from sentence count and word count and character
count.
004.      :param sent_count: integer, total number of sentences
005.      :param word_count: integer, total number of words
006.      :param char_count: integer, total number of characters
007.      :return: integer, ARI index
008.      '''
009.         return 4.71 * (char_count / word_count) + 0.5 * (word_count /
sent_count) - 21.43
010.
011.  def Flesh_Kincaid_grade_level(sentence_count, word_count, syllable_count):
012.      '''
013.      Calculate Flesh Kincaid grade level.
014.      :param sentence_count: integer, total number of sentences
015.      :param word_count: integer, total number of words
016.      :param syllable_count: integer, total number of syllables
017.      :return: integer, Flesh Kincaid grade level
018.      '''
019.      return 0.39 * (word_count / sentence_count) + 11.8 * (syllable_count /
word_count) - 15.59
020.
021.  def interpret_ARI(index):
022.      rounded_index = round(index)
023.      if rounded_index == 2 or rounded_index == 1:
024.          return 'A1'
025.      elif rounded_index == 3 or rounded_index == 4:
026.          return 'A2'
027.      elif rounded_index == 5 or rounded_index == 6 or rounded_index == 7:
028.          return 'B1'
029.      elif rounded_index == 8 or rounded_index == 9 or rounded_index == 10:
030.          return 'B2'
031.      elif rounded_index == 11 or rounded_index == 12:
032.          return 'C1'
033.      elif rounded_index == 13 or rounded_index == 14:
```

```
034.          return 'C2'
035.      else:
036.          return '?'
037.
038.  def interpret_Flesh_Kincaid_grade(index):
039.      if index < 3:
040.          return 'A1'
041.      elif index < 6:
042.          return 'A2'
043.      elif index < 9:
044.          return 'B1'
045.      elif index < 12:
046.          return 'B2'
047.      elif index < 15:
048.          return 'C1'
049.      else:
050.          return 'C2'
```

Додаток 5. Таблиця переходу грамем у рівні CEFR

рівень CEFR	Окремі формальні вимоги	Приблизні морфологічні теги-відповідники
A1	загальні родові значення іменників в однині: чоловічий рід, жіночий рід, середній рід: чоловік, жінка, місто	{'NOUN', 'sing'},
A1	утворення форм множини у називному відмінку: закінчення –и, –і (ї): інститут – інститути, лікар - лікарі, музей – музеї;	{'NOUN', 'plur', 'nomn'},
A1	- відмінювання у знахідному та місцевому відмінках іменників чоловічого роду з твердою основою: пан, син, брат, тато, Данило;	{'NOUN', 'accs', 'sing'}, {'NOUN', 'loct', 'sing'},
A1	- відмінювання у знахідному та місцевому відмінках іменників чоловічого роду з м’якою основою (кінцеві компоненти ь, й): учитель, Василь, готель, край, герой;	
A1	- відмінювання у знахідному та місцевому відмінках іменників жіночого роду з твердою основою (закінчення –а після одного приголосного): мама, сестра, Оксана, Україна, Європа, рука;	
A1	- відмінювання у знахідному та місцевому відмінках іменників жіночого роду з м’якою основою (закінчення –я після одного приголосного): бабуся, доня, Галя.	
A1	- відмінювання у знахідному та місцевому відмінках іменників жіночого роду з м’якою основою (кінцевий компонент –ія): мрія, Надія, аварія;	
A1	- відмінювання у знахідному та місцевому відмінках іменників середнього роду	

	(закінчення -о, -е): місто, морозиво, місце, серце;	
A1	- уживання форм кличного відмінка: Остапе, Олено, Надіє, Галю.	{'NOUN', 'voct'},
A1	- узгодження прикметників з іменниками в роді й числі: іноземний студент, домашній одяг; зелена сумка, літня погода; велике вікно, останнє питання; іноземні студенти, зелені сумки, великі вікна, останні питання;	{'ADJF', 'sing'}, {'ADJF', 'plur'},
A1	- відмінювання у знахідному та місцевому відмінках прикметників чоловічого роду із закінченням –ий: великий, зелений, іноземний;	{'ADJF', 'nomn'}, {'ADJF', 'accs'}, {'ADJF', 'loct'}
A1	- відмінювання прикметників жіночого роду із закінченням –а: велика, зелена, іноземна;	
A1	- відмінювання у знахідному та місцевому відмінках прикметників середнього роду із закінченням –е: велике, зелене, іноземне;	
A1	- відмінювання у знахідному та місцевому відмінках прикметників чоловічого роду із закінченням –ій: літній, домашній;	
A1	- відмінювання у знахідному та місцевому відмінках прикметників жіночого роду із закінченням –я: літня, домашня;	
A1	- відмінювання у знахідному та місцевому відмінках прикметників середнього роду із закінченням –є: літнє, домашнє;	
A1	- уживання відмінкових форм особових займенників: я (мене, мені), ти (тебе, тобі), ми (нас, нам), ви (вас, вам); утворення родових форм присвійних займенників мій, моя, моє, мої; твій, твоя, твоє, твої; наш, наша, наше, наші; ваш, ваша, ваше, ваші, його, її, їхній,	{'NPRO', 'pers'}, {'NPRO'}

	їхня, їхнє, їхні.	
A1	- утворення родових форм вказівних займенників той, та, те, ті; цей, ця, це, ці.	
A1	- дієвідмінювання дієслів у теперішньому часі (читаю, читаєш, читає, читаємо, читаєте, читають);	{'VERB', 'plur'}
A1	- дієвідмінювання дієслів у теперішньому часі (працюю, працюєш, працює, працюємо, працюєте, працюють);	
A1	- дієвідмінювання дієслів у теперішньому часі (можу, можеш, може, можемо, можете, можуть; іду, ідеш, іде, ідемо, ідете, ідуть; їду, їдеш, їде, їдемо, їдете, їдуть);	
A1	- дієвідмінювання дієслів у теперішньому часі (сиджу, сидиш, сидить, сидимо, сидите, сидять);	
A1	- дієвідмінювання дієслів у теперішньому часі (дивлюся, дивишся, дивиться, дивимося, дивитеся, дивляться);	
A1	- форми майбутнього часу деяких дієслів доконаного виду (зможу ..., скажете ..., прочитають ...)	{'VERB', 'perf', 'futr'}
A1	- форми наказового способу деяких дієслів (читай, читайте (прочитайте), скажи, скажіть, дай, дайте, пиши, пишіть (напишіть), сідай, сідайте, сиди, сидіть).	{'VERB', 'impr'}
A2	- відмінювання іменників чоловічого роду з твердою основою: чоловік, студент, батько, сусід, магазин, Львів; 32	'NOUN'
A2	- відмінювання іменників чоловічого роду з твердою основою (кінцевий компонент –ар): лікар, аптекар, базар.	

A2	- відмінювання іменників чоловічого роду з твердою основою (кінцеві компоненти ж, ч, щ, ш): викладач, товариш, дощ, вантаж;	
A2	- відмінювання іменників чоловічого роду з м’якою основою (кінцеві компоненти ь, й): Андрій, дідусь, коваль, музей, водій;	
A2	- відмінювання іменників жіночого роду з твердою основою (закінчення –а після одного приголосного): Олена, Ганна, кімната, газета;	
A2	- відмінювання іменників жіночого роду з м’якою основою (закінчення –я після одного приголосного): бабуся, Оля, Маруся.	
A2	- відмінювання іменників жіночого роду з м’якою основою (кінцевий компонент –ія): Вікторія, Марія, авдиторія, компанія, Італія;	
A2	- відмінювання іменників середнього роду із закінченням –о: око, вухо, вікно, залізо;	
A2	- відмінювання іменників середнього роду із закінченням –е: місце, поле, серце;	
A2	- відмінювання іменників середнього роду із закінченням –я: життя, обличчя, здоров’я.	
A2	- відмінювання прикметників чоловічого роду із закінченням –ий: український, високий, дешевий;	{'ADJF', 'sing'}, {'ADJF, 'plur'}
A2	- відмінювання прикметників жіночого роду із закінченням –а: українська, висока, дешева;	
A2	- відмінювання прикметників середнього роду із закінченням –е: українське, високе, дешеве;	
A2	- відмінювання прикметників чоловічого роду із закінченням –ій: синій, осінній, домашній;	
A2	- відмінювання прикметників жіночого роду із закінченням –я: синя, осіння, домашня;	

A2	- відмінювання прикметників середнього роду із закінченням –є: синє, осіннє, домашнє;	
A2	- відмінювання прикметників у формі множині –і: українські, високі, дешеві, сині, осінні, домашні.	
A2	- відмінювання родових форм порядкових числівників із закінченням –ий, -а, -е: десятий, десятого, ..., десята, десятої, ..., десяте, десятого;	
A2	- відмінювання родових форм порядкових числівників із закінченням –ій, -я, -є: третій, третього, ..., третя, третьої, ..., третє, третього;	
A2	- відмінювання родових форм кількісного числівника один: один, одного, одному, одним, на одному; одна, одної, одній	
A2	- відмінювання особових займенників: я (мене, мені, мною), ти (тебе, тобі, тобою), ми (нас, нам, нами), ви (вас, вам, вами);	{'NPRO', 'pers'}
A2	- відмінювання присвійних займенників чоловічого і середнього родів мій, моє (мого, моєму, моїм), твій, твоє (твого, твоєму, твоїм), наш, наше, ваш, ваше,	{'NPRO'}
A2	- відмінювання присвійного займенника їхній у формі чоловічого і середнього родів їхній, їхнє; незмінна форма присвійного займенника його;	
A2	- відмінювання присвійного займенника їхня у формі жіночого роду їхня, їхньої; їхній, їхньою, на їхній; незмінна форма присвійного займенника її.	
A2	- відмінювання присвійного займенника їхня у формі жіночого роду їхня, їхньої; їхній, їхньою, на їхній; незмінна форма присвійного	

	займенника її.	
A2	- дієвідмінювання дієслів у теперішньому часі (мрію, мрієш, мріє, мріємо, мрієте, мріють);	{'VERB', 'pres'}
A2	- дієвідмінювання дієслів у теперішньому часі (кажу, кажеш, каже, кажемо, кажете, кажуть; іду, ідеш, іде, ідемо, ідете, ідуть; їду, їдеш, їде, їдемо, їдете, їдуть);	
A2	- дієвідмінювання дієслів у теперішньому часі (кричу, кричиш, кричить, кричимо, кричите, кричать; ходжу, ходиш, ходить, ходимо, ходите, ходять; їжджу, їздиш, їздить, їздимо, їздите, їздять);	
A2	- дієвідмінювання дієслів у теперішньому часі (п’ю, п’єш, п’є, п’ємо, п’єте, п’ють);	
A2	- дієвідмінювання дієслова боротися в теперішньому часі (борюся, борешся, бореться, боремося, боретеся, борються);	боротися' {'VERB', 'pres'}
A2	- утворення форм минулого часу дієслова (мріяв, мріяла, мріяло, мріяли; боровся, боролася, боролося, боролися; пив, пила, пило, пили)	{'VERB', 'past'}
A2	- майбутній час дієслів недоконаного виду – складена форма (буду боротися, будеш пити, буде кричати, будемо мріяти ...);	{'VERB', 'futr'}
A2	- майбутній час дієслів доконаного виду (скажу..., закричите..., поборють...);	
A2	- проста форма вищого ступеня: солодший, важливіший;	{'ADJF', 'COMP'}
A2	- проста форма найвищого ступеня: найсолодший, найважливіший;	{'ADJF', 'Supr'}
A2	- складена форма вищого ступеня: більш солодкий, менш кислий, більш важливий,	більш', 'менш', 'більше', 'менше'

	менш важливий;	
A2	- складена форма найвищого ступеня порівняння: найбільш солодкий, найменш кислий, найбільш важливий, найменш важливий;	найбільш', 'найменш', 'найбільше', 'найменше'
A2	- суплетивні форми: великий – більший, найбільший; малий – менший, найменший.	{'ADJF', 'COMP'}, {ADJF, 'Supr'}
A2	- складнопідрядні реченні зі сполучником щоб для вираження мети того самого суб’єкта: Ми вчимо українську мову, щоб розмовляти з друзями.	
B1	- відмінювання іменників чоловічого роду з основою на твердий приголосний: чемпіон, стан, метр, холод, континент, Лев, Валентин, Лондон;	{'NOUN'}
B1	- відмінювання іменників чоловічого роду з твердою основою (кінцевий компонент –ар): пекар, комісар, узвар, гектар.	
B1	- відмінювання іменників чоловічого роду з кінцевим компонентом –яр: футляр, перпендикуляр; школяр, пісняр, каменяр.	
B1	- відмінювання іменників чоловічого роду з кінцевим компонентом –о, у тому числі власні назви: Шевченко, Андрійко, Махно, безхатько;	
B1	- відмінювання іменників чоловічого роду з кінцевими компонентами ж, ч, щ, ш, у тому числі власні назви: сторож, викладач, слухач, товариш, Свищ, Драч;	
B1	- відмінювання іменників чоловічого роду з м’якою основою (кінцеві компоненти ь, й): кінь, велетень, Кудлай;	
B1	- відмінювання іменників чоловічого роду з кінцевими компонентами -ець, - ень: хлопець,	

	червень, українець, пень;	
B1	- відмінювання іменників чоловічого роду з кінцевими компонентами -ин (- анин, -янин): громадянин, болгарин, селянин, киянин, вінничанин;	
B1	- відмінювання іменників чоловічого роду з кінцевим компонентом -ий (прикметникове походження): Коцюбинський, вартовий, молодий;	
B1	- відмінювання іменників жіночого роду з твердою основою (закінчення -а): Ольга, каса, касирка, донечка, жінка, студентка, записка;	
B1	- відмінювання іменників жіночого роду з м’якою основою (закінчення -я): робітниця, Катруся, парасоля, стаття.	
B1	- відмінювання іменників жіночого роду з м’якою основою (кінцевий компонент -ія): Анастасія, філія, Іспанія, сім’я;	
B1	- відмінювання іменників жіночого роду із нульовим закінченням: осінь, тінь, любов, подорож.	
B1	- відмінювання іменників жіночого роду із кінцевим компонентом -ість: радість, молодість.	
B1	- відмінювання іменників середнього роду із закінченням –о: відро, дитятко, гроно;	
B1	- відмінювання іменників середнього роду із закінченням –е: явище, озерце, середовище;	
B1	- відмінювання іменників середнього роду із закінченням –я: завдання, знаряддя, прислів’я.	
B1	- відмінювання множинних іменників: Карпати, Черкаси, двері, окуляри.	

B1	- відмінювання прикметників чоловічого, жіночого, середнього роду із закінченнями -ий, -а, -е: вагомий, вагома, вагоме; відомий, відоме, відоме; демократичний, демократична, демократичне;	{'ADJF'}
B1	- відмінювання прикметників чоловічого, жіночого, середнього роду із закінченнями -ій, -я, -є: всесвітній, всесвітня, всесвітнє; сьогоднішній, сьогоднішня, сьогоднішнє;	
B1	- форми прикметників чоловічого роду із нульовим закінченням у називному та знахідному відмінках: зелен, молод, потрібен;	
B1	- відмінювання прикметників у формі множини -і: європейські, червоні, важливі, лінійні.	
B1	- відмінювання кількісних числівників один (одна, одне, одні), два (дві), три, чотири: один, одного, двох, двом, на трьох; чотирьох, чотирма ...;	один', 'два', 'три', 'чотири', {'NUMR'}
B1	- відмінювання неозначених числівників: багато – багатьох, багатьма, на багатьох; кілька – кількох, кількома, на кількох; декілька – декількох, декільком, декількома...	багато', 'кілька', 'декілька'
B1	- відмінювання особових займенників: я (мене, мені, мною), ти (тебе, тобі, тобою), ми (нас, нам, нами), ви (вас, вам, вами);	{'NPRO'}
B1	- відмінювання присвійних займенників чоловічого, середнього і жіночого родів: мій, моє (мого, моєму, моїм), моя (моєї, моїй, моєю); твій, твоє (твого, твоєму, твоїм), твоя (твоєї, твоїй, твоєю); наш, наше, наша; ваш, ваше, ваша;,	
B1	- відмінювання присвійного займенника їхній у формі чоловічого і середнього родів їхній, їхнє;	

	незмінна форма присвійного займенника його;	
B1	- відмінювання присвійного займенника їхня у формі жіночого роду їхня, їхньої; їхній, їхньою, на їхній; незмінна форма присвійного займенника її;	
B1	- відмінювання особово-вказівних займенників: він (його, у нього, йому, ним, на ньому), вона (її, біля неї, їй, нею, на ній), воно (його, у нього, йому, ним, на ньому), вони (їх, для них, ним, ними, на них).	
B1	- відмінювання зворотного займенника себе (собі, собою, на собі).	себе', 'соб'
B1	- відмінювання вказівних займенників той, цей.	той', 'цей'
B1	- дієвідмінювання дієслів у теперішньому часі (запрошую, запрошуєш, запрошує, запрошуємо, запрошуєте, запрошують, мию(ся), миєш(ся), миє(ся), миємо(ся), миєте(ся), миють(ся));	{'VERB', 'Refl', 'pres'}
B1	- дієвідмінювання дієслів дати, їсти у теперішньому часі (даси, дамо, ... дадуть; їм, їси, їсть, їмо, їсте, їдять);	дати', 'їсти' {'VERB', 'pres'}
B1	- утворення форм роду і числа дієслів минулого часу недоконаного і доконаного виду (хотів, хотіла, хотіло, хотіли; побачив, побачила, побачило, побачили);	{'VERB', 'past', 'impf'}, {'VERB', 'past', 'perf'}
B1	- дієвідмінювання дієслів недоконаного виду в майбутньому часі – аналітична форма (боротимуся, питимеш, кричатиме, любитимете, мріятимуть ...);	{'VERB', 'futr', 'impf'}
B1	- дієвідмінювання дієслів доконаного виду в майбутньому часі (скажу ..., закричите ..., поборють ...).	{'VERB', 'futr', 'perf'}
B1	- утворення форм наказового способу (мрій,	{'VERB' 'impr'} 'хай', 'нехай',

	кажіть (скажіть), пий, пийте, борися, боріться, хай/нехай бореться, хай/нехай борються).	'нехаяти'
B1	- утворення форм умовного способу (говорив би, випила б, прочитали б).	би', 'б'
B1	а) проста форма вищого ступеня: важкий – важчий, легкий – легший, дорогий – дорожчий;	{'ADJF' 'COMP'}
B1	б) проста форма найвищого ступеня: найважчий, найлегший, найдорожчий;	{'ADJF' 'Supr'}
B1	в) складена форма вищого ступеня: більш синій, менш холодний, більш дорогий;	більш', 'менш', 'більше', 'менше'
B1	г) складена форма найвищого ступеня порівняння: найбільш високий, найменш потрібний;	найбільш', 'найменш', 'найбільше', 'найменше'
B1	а) проста форма вищого ступеня: важко – важче, легко – легше;	{'ADVB' 'COMP'}
B1	б) проста форма найвищого ступеня: найважче, найлегше;	{'ADVB' 'Supr'}
B1	в) складена форма вищого ступеня: більш сучасно, менш холодно, більш важливо;	більш', 'менш', 'більше', 'менше'
B1	г) складена форма найвищого ступеня порівняння: найбільш високо, найменш потрібно;	найбільш', 'найменш', 'найбільше', 'найменше'
B2	- відмінювання іменників чоловічого роду з основою на твердий приголосний, у тому числі власні назви: соняшник, каталог, колектив, Кривий Ріг, Близький Схід;	{'NOUN'}
B2	- відмінювання іменників чоловічого роду з твердою основою (кінцевий компонент –ар/-яр), у тому числі власні назви: календар, Кот-д'Івуар, парламентар, ліхтар, Катар, скляр, газетяр, Дігтяр;	
B2	- відмінювання іменників чоловічого роду з	

	кінцевим компонентом –яр, у тому числі власні назви: скляр, газетяр, футляр, перпендикуляр; школяр, пісняр, каменяр.	
B2	- відмінювання іменників чоловічого роду із закінченнями -о, -а у тому числі власні назви: Дніпро, Петренко, Франко, воєвода, Микола, Полюга, Петлюра, Сковорода;	
B2	- відмінювання іменників чоловічого роду з кінцевими компонентами ж, ч, щ, ш, у тому числі власні назви: хрущ, Балаш, Гадяч, вуж;	
B2	- відмінювання іменників чоловічого роду з м’якою основою (кінцевий компонент -ьо): дідуньо, татуньо, Бенедьо;	
B2	- відмінювання іменників чоловічого роду з кінцевими компонентами -ець, - ень: день, травень, знавець;	
B2	- відмінювання іменників назв осіб чоловічого роду з кінцевим компонентом - ин (-анин, -янин): татарин, освітянин, черкащанин;	
B2	- відмінювання іменників чоловічого роду з кінцевим компонентом -ий (прикметникове походження): Котляревський, Кропивницький;	
B2	- відмінювання іменників жіночого роду із закінченням -а/-я, -ія: хата, вишня, армія, Саудівська Аравія;	
B2	- відмінювання іменників жіночого роду із закінченням -а (прикметникове походження): набережна, наречена, учительська, Заньковецька, Лозова (місто);	
B2	- відмінювання іменників жіночого роду із нульовим закінченням: Прип’ять, сіль, мазь, матір;	
B2	- відмінювання іменників середнього роду із	

	закінченнями -о, -е: скло, паливо, прізвище;	
B2	- відмінювання іменників середнього роду із закінченням -а/-я (із появою суфіксів -ат/-ят, -ен): курча, немовля, плем’я, ім’я;	
B2	- відмінювання іменників середнього роду із закінченням -е/-є (прикметникове походження), у томі числі власні назви: пальне, майбутнє, минуле, Рівне;	
B2	- відмінювання множинних іменників: ножиці, гроші, сутінки, шахи, Суми, Чернівці;	
B2	- відмінювання іменників спільного роду: листоноша, приблуда, базікало, плакса;	
B2	- відмінювання абревіатур та складноскорочених слів: ЦУМ – ЦУМом, у ЦУМі; Дніпрогес – Дніпрогесу, Дніпрогесом, на Дніпрогесі;	{'NOUN' 'Abbr'}
B2	-відмінювання прикметників чоловічого, жіночого, середнього роду із закінченнями -ий, -а, -е: народний, народна, народне; багатий, багата, багате; важливий, важлива, важливе;	{'ADJF'}
B2	-відмінювання прикметників чоловічого, жіночого, середнього роду із закінченнями -ій, -я, -є: осінній, осіння, осіннє, могутній, могутня, могутнє;	{'ADJF'}
B2	- відмінювання прикметників чоловічого роду із нульовим закінченням: варт, сестрин, братів, Андріїв;	{'ADJF'}
B2	-відмінювання складних прикметників із компонентом -лиций: білолиций, білолицього..., круглолиця, круглолицьої...;	{'ADJF'}
B2	-відмінювання прикметників у формі множини (закінчення –і): київські, братові, міжнародні,	{'ADJF'}

	сестрині, материнські;	
B2	-відмінювання складних прикметників із першим компонентом кількісним числівником: двох’ярусний, трикутний, двадцятип’ятирічний;	{'ADJF'}
B2	- відмінювання присвійного займенника їхній, їхня, їхнє, їхні;	їхній'
B2	- відмінювання зворотного займенника себе (собі, собою, на собі).	себе', 'соб'
B2	- дієвідмінювання дієслів недоконаного виду в теперішньому часі (вимірюю, міряєш, передає, зостаємося ..., везу, везеш, везе, веземо, везете, везуть);	{'VERB' 'impf', 'pres'}
B2	- дієвідмінювання дієслів недоконаного виду в майбутньому часі (буду висловлювати = висловлюватиму, висловлюватимеш, висловлюватиме, висловлюватимемо, ... висловлюватимуть);	{'VERB' 'impf', 'futr'}
B2	- дієвідмінювання дієслів дати (роздати, передати), їсти (з’їсти) у майбутньому і теперішньому часі (даси, дамо, ... дадуть; їм, їси, їсть, їмо, їсте, їдять);	дати', 'їсти', {'VERB', 'pres'}, {'VERB', 'futr'}
B2	- змінювання дієслів недоконаного і доконаного виду в минулому часі за родами і числами (кликав, кликала, кликало, кликали, ... облетів, облетіла, облетіло, облетіли);	
B2	- дієвідмінювання дієслів доконаного виду в майбутньому часі (звикну, звикнеш..., присвячу, присвятиш, ... присвятять, прожену, проженеш, прожене, проженемо...)	{'VERB' 'perf'}

B2	- утворення форм наказового способу (приходь, приходьте, літай, літайте, літаймо, хай/нехай літає(-ють);	{'VERB' 'impr'} 'хай', 'нехай', 'нехаяти'
B2	- утворення форм умовного способу (знав би, придумала б, відслужили б, якби закричав, ла, ло, ли);	би', 'б'
B2	а) проста форма вищого ступеня: молодий – молодший, солодкий – солодший, сильний – сильніший;	{'ADJF' 'COMP'}
B2	б) проста форма найвищого ступеня: наймолодший, найсолодший, найсильніший;	{'ADJF' 'Supr'}
B2	в) складена форма вищого ступеня: більш/менш солодкий, більш/менш сильний;	більш', 'менш', 'більше', 'менше'
B2	г) складена форма найвищого ступеня порівняння: найбільш/найменш солодкий; ґ) суплетивні форми: великий – більший, малий – менший.	найбільш', 'найменш', 'найбільше', 'найменше'
B2	а) проста форма вищого ступеня: солодко – солодше, сильно – сильніше, доступно – доступніше;	{'ADVB' 'COMP'}
B2	б) проста форма найвищого ступеня: найсильніше, найдоступніше;	{'ADVB' 'Supr'}
B2	в) складена форма вищого ступеня: більш/менш корисно, більш/менш доступно; г) складена форма найвищого ступеня порівняння: найбільш/найменш корисно; ґ) суплетивні форми: багато – більше, мало – менше.	більш', 'менш', 'більше', 'менше', 'найбільш', 'найменш', 'найбільше', 'найменше'
B2	- складнопідрядні речення з відношеннями умови:	якщо', 'хоч', 'якби'
C1	- відмінювання іменників чоловічого роду з основою на твердий приголосний, у тому числі власних назв: магніт, злак, фактаж, Алжир	{'NOUN'}

	(місто), Алжир (країна), буквар, якір;	
C1	- відмінювання іменників чоловічого роду з основою на м’який приголосний, у тому числі власних назв: тунель, корінь, шампунь, урожай, Хмельницький, Ковель;	
C1	- відмінювання іменників жіночого роду: сльоза, фраза, Катерина Петрівна, крапля, гривня, кишеня, ніздря, сажа, відсіч, латинь, Рось;	
C1	- відмінювання іменників середнього роду: сузір’я, листя, збентеження, місяченько, щастя, колесо, Поділля;	
C1	- відмінювання складних іменників, у тому числі з другою частиною прикметникового походження: лісостеп, хліб-сіль, Австро-Угорщина, Кам’янець-Подільський, Рава-Руська;	
C1	- відмінювання множинних іменників: пестощі, солодощі, оглядини, вершки, шашки, Прилуки;	
C1	- відмінювання іменників спільного роду: сирота, базіка, шульга, ледащо, каліка;	
C1	- відмінювання абревіатур та складноскорочених слів: ЛАЗ (ЛАЗу, на ЛАЗі), райспоживспілка (райспоживспілки, райспоживспілці), Мін’юст (для Мін’юсту, у Мін’юсті);	{'NOUN' 'Abbr'}
C1	- відмінювання прикметників чоловічого, жіночого, середнього роду із закінченнями -ий, -а, -е: щасливий, а, е; природний, а, е; незліченний, а, е; розкішний, а, е; рівноправний, а, е;	{'ADJF'}
C1	- відмінювання прикметників чоловічого, жіночого, середнього роду із закінченнями -ій,	

	-я, -є: дружній, я, є; тутешній, я, є, довгошиїй, -я, є, могутній, -я, -є, а також чол. роду – орлій;	
C1	- відмінювання прикметників чоловічого роду із нульовим закінченням: Миколин, тітчин, офіцерів, Галин, Ігорів, Маріїн, смутен, годен;	
C1	- відмінювання складних прикметників: волелюбний, синьо-жовтий, гіркувато-солоний, хвилеподібний; білолиций, круглолиций;	
C1	- відмінювання прикметників у формі множини (закінчення –і): лінгвістичні, відсутні, самобутні, далекії, заморські;	
C1	- відмінювання зворотного займенника себе (собі, собою, на собі).	себе', 'соб'
C1	1.1.8.1 Ступені порівняння якісних прикметників:	
C1	а) проста форма вищого і найвищого ступенів: відомий – відоміший – найвідоміший; рухливий – рухливіший – найрухливіший; глибокий – глибший – найглибший;	{'ADJF' 'COMP'} {'ADJF' 'Supr'}
C1	б) складена форма вищого і найвищого ступенів: більш/менш відомий; найбільш/найменш відомий;	більш', 'менш', 'більше', 'менше', 'найбільш', 'найменш', 'найбільше', 'найменше'
C2	- відмінювання іменників чоловічого роду з основою на твердий приголосний, у тому числі власних назв: договір, символ, тамбур, лемко, Бойко, Харків, Хотин, вельможа;	{'NOUN'}
C2	- відмінювання іменників чоловічого роду з основою на м’який приголосний, у тому числі власних назв: трофей, гість, ступінь, бородань, ковбой, кий, Михась, Ізраїль, Дунай;	
C2	- відмінювання іменників жіночого роду: прем’єра, королівна, русявка, Ярка, Ярця, Яринка, мелодія, оселя, далеч, станція Пісочна,	

	Варшава, Кіцмань;	
C2	- відмінювання іменників середнього роду: словникарство, плесо, горище, верб’я, Опілля, рішення, гусеня, собача;	
C2	- відмінювання складних іменників усіх родів, у тому числі з частиною прикметникового, числівникового чи дієслівного походження: чорнослив, верболіз, прем’єр-міністр, бетоновоз, Лисичка-Сестричка, самохід, водолікування, велодоріжка, чотирикутник;	
C2	- відмінювання множинних іменників: шорти, надра, дрова, прийми, перегони, Піренеї, ворота;	
C2	- відмінювання іменників спільного роду: волоцюга, сіромаха, бездара, невдаха, зірвиголова, зубрило;	
C2	- відмінювання абревіатур та складноскорочених слів: ДАК, неп, Кабмін, медпрацівник, завгосп, профспілка;	
C2	- відмінювання прикметників чоловічого, жіночого, середнього роду із закінченнями -ий, -а, -е: найтяжчий, а, е; світовий, а, е; священний, а, е; безробітний, а, е; вороний, а, е;	{'ADJF'}
C2	- відмінювання прикметників чоловічого, жіночого, середнього роду із закінченнями -ій(-їй), -я, -є: внутрішній, я, є; досвітній, я, є, безкраїй, я, є, горішній, я, є, довгошиїй, я, є;	
C2	- відмінювання прикметників чоловічого роду із нульовим закінченням: Михайлів, воротарів, ювілярів, Андріїв, бабин, світел, благословен;	
C2	- відмінювання складних прикметників: повнолітній, яскраво-зелений, історико-	

	культурний, важкохворий; повнолиций, смуглолиций;	
C2	- відмінювання особових займенників: я (мене, мені, мною), ти (тебе, тобі, тобою), ми (нас, нам, нами), ви (вас, вам, вами); він (його, у нього, йому, ним, на ньому), вона (її, біля неї, їй, нею, на ній), воно (його, у нього, йому, ним, на ньому), вони (їх, для них, ним, ними, на них).	{'NPRO'}
C2	- відмінювання присвійних займенників: мій, моя, моє, мої; твій, твоя, твоє, твої; свій, своя, своє, свої; наш, наша, наше, наші; ваш, ваша, ваше, ваші; їхній, їхня, їхнє, їхні;	
C2	- відмінювання зворотного займенника себе (собі, собою, на собі).	себе', 'соб'